

# The Modified Finite Particle <br> Method in the context of meshless methods 

by<br>Andrea Montanino<br>A Thesis submitted in Fulfilment of the Requirements for the Degree of Doctor in Philosophy in<br>Computational Mechanics and Advanced Materials at Istituto Universitario di Studi Superiori, Pavia, Italy<br>Ph.D. Program Coordinator: Prof. F. Auricchio<br>Supervisor:<br>Prof. F. Auricchio - Università degli Studi di Pavia Coadvisors:<br>Prof. A. Reali - Università degli Studi di Pavia<br>Dr. D. Asprone - Università degli Studi di Napoli "Federico II"



A mio fratello Sergio per esserci sempre stato durante questi anni


## Acknowledgments

Non so se esista una prassi o una regola che disciplini la lingua in cui scrivere i ringraziamenti di una tesi di dottorato. In ogni caso, qualora dovesse esistere, di seguito si ricorre a una specie di deroga, dal momento che le cose che sto per scrivere saranno un puro flusso di coscienza attraverso questi ultimi quattro anni di vita, che hanno rappresentato un'avventura davvero inaspettata.

Ad ogni fine di un percorso è abitudine voltarsi indietro per valutare e guardare con sguardo critico quello che si è fatto, e questo porta a ripensare alle persone e alle situazioni che hanno contribuito a qualsiasi livello al percorso stesso. In questo caso il lavoro che si conclude è frutto certamente dell'impegno e del significativo apporto dei docenti e dei colleghi del gruppo di Meccanica Computazionale dell'Univeristà di Pavia, ma io considero ugualmente significativo e necessario il contributo umano che è venuto da tante altre persone che ho avuto la possibilità e il piacere di conoscere durante questi anni.

Ringrazio il prof. Ferdinando Auricchio, relatore di questa tesi per avermi dato la possibilità di crescere e di formarmi all'interno di un gruppo di ricerca stimolante e dinamico come quello di Meccanica Computazionale, così ricco di competenze e così aperto alla cooperazione con l'esterno, sia dal punto di vista della collaborazione universitaria, che dal punto di vista dell'apertura al mondo delle aziende.

Grazie al prof. Alessandro Reali per i suoi suggerimenti e la sua disponibilità durante questi anni, per le stimolanti chiacchierate e per i numerosi consigli. Grazie all'ing. Domenico Asprone per il suo supporto alle attività svolte durante questo dottorato.

Grazie ai tanti studenti di Ingegneria Civile dell'Università di Pavia, del corso di Meccanica Computazionale, con i quali ho potuto sperimentare quanto sia bella e stimolante l'attività didattica.

Un ringraziamento affettuoso al prof. Maurizio Angelillo, regista occulto di questo percorso di dottorato, per avermi catapultato con entusiasmo e tanta fiducia in questa esperienza di studio fuori dalla mia università di origine. Grazie anche per il sostegno ricevuto in questi anni dal punto di vista scientifico e umano, per la stima dimostratami in tante occasioni.

Questo momento è particolarmente gradito per ringraziare tante persone che hanno consentito che la mia prima esperienza fuori casa diventasse un'esperienza serena e divertente. Grazie ai tanti amici con i quali ho condiviso tante esperienze all'esterno dell'ambiente di lavoro, e tante mangiate, e tanto altro. Grazie a Carmelo per i tanti momenti di serenità e rilassatezza nella vita di coinquilini. Grazie a Lolli e Valentina, a Lorenzo e Roberta per tante giornate che abbiamo passato insieme, per la vostra presenza, la vostra amicizia.

Grazie a Carlo perché le origini non si dimenticano, e perché una parola in napoletano
(due, tre, mille... tutta una serata) ti fa sentire a casa anche a 800 km di distanza. Perché ci si capisce senza dovrsi esprimere per forza con frasi di senso compiuto.

Grazie ai numerosi colleghi del dipartimento, passati, presenti e futuri: Xi, Alice, Margherita, Tuong, Marco P., Paolo, Giulia, Anna, Adrien, Eli, Chiara P., Sonia. Grazie ad Alberto per la fiducia che hai avuto in me, e che forse continui ad avere. Grazie ad Elisa e Chiara, che oggi concludono insieme a me questo percorso, per aver condiviso i momenti e le ansie finali. Grazie Isabella, per il tuo affetto, la tua pazienza, la tua disponibilità in tante occasioni, soprattutto perché finalmente hanno trovato chi prendere in giro dopo che hanno finito con me. Grazie al resto del comitato di quartiere TTF (Marco, Elena e Rodrigo) per aver dato origine a una nuova amicizia e nuove tradizioni (che finiscono sempre a mangiare, chissà come mai): la vostra amicizia è stata una piacevole sorpresa dell'ultima ora. Grazie a Gianluca perché quando c'è da sfottere qualcuno lui è sempre presente, grazie a Stefania per avermi lodevolmente supportato in questa missione, e per continuare a farlo. Grazie a Julia e Josef per avere organizzato tante occasioni di compagnia. Grazie a Pinuccio per i numerosi, ... pacati... scambi di idee sulle nostre attività, e per i tanti consigli su come districarmi in certe situazioni extrauniversitarie. Grazie Carolina per il tuo aiuto in ogni occasione. Grazie ad Armando perché la sopporti...

Un ringraziamento speciale a Mauro che ha condiviso con me il dottorato dal suo inizio e che almeno nelle intenzioni è stato concluso insieme, nelle intenzioni e a livello di festeggiamenti. Mi aspetto che la conclusione del mio dottorato sia celebrato nelle stesse modalità, con la tua presenza.

Grazie a Michele e Simone, avete svolto il ruolo di fratelli maggiori di questo periodo della mia vita, grazie per i vostri consigli, e soprattutto per l'accoglienza che mi avete riservato nei primi momenti dal trasferimento. Grazie a voi sono riuscito presto ad ambientarmi nella nuova vita a Pavia, e per me è stata una cosa molto importante. Grazie a Sissi per sopportare Simone, certo, ma soprattutto per il tuo affetto sincero e la disponibilità. Grazie a Giovannella, perchè praticamente Benevenuti al Sud e Totò e Peppino a Milano non sono niente al confronto delle nostre chiacchierate.

Grazie a Simone e Pasquale, emigranti salernitani a Milano, che hanno permesso periodicamente di ricostruire un ambiente familiare in questa terra di nebbia (vabè, almeno una volta la nebbia la dovevo nominare).

Le amicizie sviluppate in questi anni ovviamente non cancellano quelle costruite in tanti anni di vita e di studio a Salerno. Grazie a tutta la comunità della parrocchia di San Pietro in Camerellis a Salerno, per riaccogliermi prontamente a ogni ritorno, per farmi sentire parte integrante di essa anche se la mia vita si svolge per oltre 300 giorni all'anno lontano. Grazie a don Francesco, ai gruppi giovani ciacianielli, giovani down, diversamente giovani (come si chiama?) e alle singole persone che li compongono.

Grazie a puffo Capoccia e allo spartano che fu, e grazie a Mario perchè vedi, alla fine ci sei sempre, anche se per la prima volta abbiamo separato le nostre strade dopo 18 anni di vita continuati alla stessa parte.

Grazie ai ragazzi dell'università di Salerno per l'amicizia che è rimasta in piedi nei diversi percorsi che abbiamo intrapreso, in particolare grazie a Fabrizio, perchè le pippe non sono finite nonostante tutto, grazie a Giulio, Maurizio, Alessandro, Enza, Simona, Gerardo e Arianna.... ah, e Federico.

Grazie a mammina e papuccio, che hanno sempre incoraggiato con fiducia questa mia
esperienza, e non hanno mai fatto mancare il loro sostegno e la piena condivisione delle scelte. Grazie perché come è sempre stato, anche in questa occasione avete fatto prevalere la possibilità di una buona opportunità di crescita per me sulla normalità e tranquillità della vita quotidiana vicino. Grazie per le tante cose dette, per le tante che siete riusciti a non dire...

Grazie infine a mio fratello Sergio, che ha svolto la missione più importante, quella cioè di accompagnarmi, consigliarmi, guidarmi saggiamente nel nuovo percorso, essermi sempre vicino nei momenti di difficoltà che pure ci sono stati.

Alla fine di questo dottorato praticamente non so quello che sarà di me nel prossimo futuro. Certamente questa esperienza mi ha fatto superare dei muri che quattro anni fa non avrei mai pensato di riuscire a saltare. Adesso posso guardare davanti a me con maggiore consapevolezza.... per il resto, con l'aiuto di tante persone, spero di individuare quale sia la strada che veramente voglio seguire.

Andrea

## Abstract

Meshless methods are nowadays widely investigated in the framework of numerical methods due to their characteristic of being free of any predetermined connectivity among nodes, such as a grid or a mesh. This characteristic makes such methods particularly flexible in problems where large deformations are involved, such as high-velocity impact, metal foaming, and fluiddynamics.

In the present thesis we report the development of a new numerical method, called Modified Finite Particle Method, from its introduction in Asprone et al. (2010) until its most recent developments. The Modified Finite Particle Method (MFPM) is a numerical approximation technique of functions and derivatives inspired by the Modified Smoothed Particle Hydrodynamics (MSPH), a numerical algorithm belonging to the class of the SPH-derived methods.

The first development of the Modified Finite Particle Method from its original formulation has given the method higher computational efficiency by removing the need of numerical integration, and hence, tessellation of the domain; such a novel formulation is then applied to 1D and 2D elasticity and has been checked in terms of computational efficiency and numerical error (Asprone et al., 2014). Then the method is extended to incompressible materials, in particular we explore the numerical difficulties connected to a quasi-incompressible material when using a displacement-based formulation.

At a later stage we approach the solution of the Stokes equations, that model full incompressible materials: such equations are the topic of many scientific works due to the existence of a numerical limitation known as the inf-sup or $L B B$ condition, which imposes restrictions in the discretization of displacement and pressure fields. In the field of collocation methods such as the Finite Difference Method, as an example, staggered grids are used; unfortunately such strategy is not extensible to meshless methods, which in general deal with non-structured distributions of collocation points. Hence alternative formulations of the Stokes problem have been discussed and investigated using the Modified Finite Particle Method, obtaining the expected accuracy in terms of error convergence.

In the last part of the thesis we develop an extended formulation of the Modified Finite Particle Method, consisting in a combination of the MFPM with a Least Square Residual Method, and apply it to Stokes and Navier-Stokes equations. Such an algorithm permits to solve the original formulation of Stokes and Navier-Stokes problems using the same discretization for velocity and displacement, overcoming the limitations given by the LBB condition, and without the need of alternative continuous formulations. Moreover the algorithm looks more robust with respect to extremely unstructured collocation point distributions.

## Contents

1 Introduction ..... 1
1.1 The Smoothed Particle Hydrodynamics ..... 2
1.2 Alternative approaches to SPH ..... 4
1.2.1 Reproducing Kernel Particle Method ..... 4
1.2.2 Corrective Smoothed Particle Method ..... 5
1.2.3 Modified Smoothed Particle Hydrodynamics ..... 6
1.3 Meshless methods based on shape functions ..... 6
1.3.1 Meshless Local Petrov-Galerkin ..... 6
1.3.2 Meshless methods based on radial basis functions ..... 7
1.3.3 The local max-ent approach ..... 10
1.4 Meshless methods based on the Taylor series ..... 15
1.4.1 Generalized Finite Difference Method ..... 15
2 Modified Finite particle Method: 1d formulation ..... 17
2.1 Modified Finite Particle Method - Derivative approximation ..... 17
2.1.1 Consideration about the smoothing function ..... 18
2.2 Novel formulation ..... 19
2.2.1 Discrete form of a 1D boundary value problem using the Modified Finite Particle Method ..... 19
2.3 Applications to 1D boundary value problems ..... 21
2.3.1 Comparison with the Finite Difference Method ..... 25
3 Modified Finite Particle Method: multidimensional formulation ..... 27
3.1 Original formulation ..... 27
3.2 Novel formulation ..... 29
3.2.1 Projection functions and supporting nodes ..... 33
3.3 Comparison with existing methods ..... 34
3.3.1 Considerations about the stencil of the derivatives ..... 34
3.4 Searching algorithm of neighbour particles ..... 35
4 Applications to elasticity ..... 39
4.1 Linear elasticity ..... 39
4.2 Numerical tests ..... 42
4.2.1 Plate with a circular hole ..... 42
4.2.2 The Cook's membrane ..... 43
4.2.3 Multi-material problems ..... 46
4.2.4 Three-dimensional elasticity problem ..... 48
4.3 Applications of the MFPM in dynamics ..... 51
4.4 1d dynamics: a bar under quasi-impulsive axial load ..... 53
4.4.1 Fourier analysis of the wave propagation problem ..... 53
4.4.2 Solution of the eigenvalue problem for the clamped bar ..... 54
4.4.3 Approximation of the dispersion relation ..... 55
4.5 2D dynamics ..... 57
4.5.1 Dynamics of a 2D bar under quasi-impulsive load ..... 58
4.5.2 Quarter of annulus under a sinusoidal body load ..... 61
5 Quasi-incompressible materials ..... 63
5.1 Governing equations ..... 63
5.2 Classical approaches for incompressibility ..... 65
5.3 Applications ..... 67
5.3.1 Square clamped on two edges under a vertical body load ..... 67
5.4 Conclusions ..... 68
6 Incompressibility ..... 71
6.1 Stokes problem: classical formulation ..... 71
6.1.1 Classical numerical schemes for the solution of the Stokes problem ..... 72
6.2 Stokes problem: alternative formulations ..... 73
6.2.1 Stokes equations in the Pressure Poisson form ..... 73
6.2.2 A pseudo-compressibility formulation of the Stokes problem ..... 74
6.2.3 The gauge method ..... 75
6.2.4 Summary of the formulations applied in the numerical tests ..... 77
6.3 Application of the MFPM to steady Stokes problems ..... 77
6.3.1 The lid-driven cavity ..... 77
6.3.2 Square with polynomial exact solution ..... 78
6.3.3 Quarter of annulus under body load ..... 82
6.3.4 Square clamped on two edges under a vertical body load ..... 85
6.4 Non-stationary Stokes problem: continous formulations ..... 88
6.4.1 The original Chorin algorithm ..... 88
6.4.2 Developments of the Chorin algorithm ..... 89
6.4.3 Gauge method in dynamics ..... 90
6.5 Application of the MFPM to a non-stationary Stokes problems ..... 91
6.5.1 Solution using algorithms D1, D2 and D3 ..... 91
6.5.2 Solution using algorithm D4 ..... 93
7 Extended Modified Particle Method ..... 97
7.1 Modified Finite Particle Method: the extended formulation ..... 97
7.2 Governing equations for incompressible flows ..... 100
7.2.1 Solution of the Stokes equations using the Modified Finite Particle Method and the Least Square Residual Method ..... 101
7.2.2 Choice of the weights ..... 103
7.3 Solution of the Stokes problem ..... 104
7.3.1 Square with known analytical solution ..... 104
7.3.2 Quarter of annulus under body load ..... 109
7.4 Navier-Stokes Equations ..... 111
7.4.1 Navier-Stokes equations and Least Square Residual Method ..... 114
7.5 Solution of the Navier Stokes problem using MFPM combined with LSRM ..... 115
7.5.1 Flow in a square domain ..... 115
7.5.2 The lid-driven cavity ..... 116
8 Conclusions ..... 119
Bibliography ..... 121

## Chapter 1

## Introduction to the thesis

In recent years meshless methods have acquired significant importance in the framework of numerical methods due to their characteristic of being free of any kind of grid or mesh, and therefore, to exhibit an higher flexibility with respect to traditional mesh-based or grid-based methods, above all the Finite Element Method (FEM) and the Finite Difference Method (FDM). The most attractive characteristic of meshless methods is the capability of easily modelling problems where large deformations occur, following a Lagrangian point of view. Among these problems, in the literature are widely investigated problems of solid large deformations, metal foaming, and fluid dynamics.

A meshless method is characterized by the fact that nodes are not given any kind of $a$ priori connectivity. Relation among nodes are then given, at each time step, on the basis of the current position among nodes. This fact one hand gives the method high flexibility in problems where particles are repeatedly muddled up, on the other hand requires that at each time step the connectivity is set up, requiring a computational cost that sometimes is comparable with the remeshing time required by mesh-based numerical methods.

In the field of meshless methods an important distinction is made between meshless particle methods and meshless "non-particle" method. In the first case, each node is characterized by a mass, a velocity and an energy, and the interaction among nodes occurs through mutual forces, according to the principle of action and reaction. In meshless "non-particle" methods, differently, nodes are simply considered as collocation points, and used as base for the discretization of function and derivatives.

Table 1.1: Classification of some numerical approximation methods

|  | Particle | Non particle |
| :--- | :--- | :--- |
| Meshless | Smoothed Particle <br> Hydrodynamics (SPH), <br> Discrete Element Method (DEM) | Radial basis collocation, <br> Discrete Least Square Residual Method |
| Mesh-based | Particle FEM | Finite Element Method (FEM) |

### 1.1 The Smoothed Particle Hydrodynamics

The first meshless particle method introduced in the literature is the Smoothed Particle Hydrodynamics (SPH), proposed by Lucy (1977) and Gingold and Monaghan (1977) for the study of astrophysical problems. The numerical approximation technique is based on the Dirac Delta equivalence

$$
\begin{equation*}
f\left(x_{i}\right)=\int_{-\infty}^{+\infty} f(x) \delta\left(x-x_{i}\right) d x \tag{1.1}
\end{equation*}
$$

where $\delta(x)$ is the Dirac Delta distribution, which has two main properties:

$$
\begin{gather*}
\delta\left(x-x_{i}\right)=0 \quad \text { if } \quad x \neq x_{i}  \tag{1.2a}\\
\int_{-\infty}^{+\infty} \delta\left(x-x_{i}\right) d x=1 \tag{1.2b}
\end{gather*}
$$

Equation (1.1) can be seen as the projection of the function $f(x)$ on the basis $\delta\left(x-x_{i}\right)$. This particular way of considering Equation (1.1) is the starting point of many numerical methods, derived from SPH formulation. Since the Dirac Delta is difficult to manage from a numerical point of view, SPH introduces an approximation of Equation (1.1), substituting the Dirac Delta distribution with a smooth function $W\left(x-x_{i}, h\right)$, called kernel function, where $h$ is the smoothing length, which defines the region $\Omega_{i}=\left[x_{i}-h, x_{i}+h\right]$ in which the smoothing function is non-zero. Consequently, Equation (1.1) is approximated through the kernel evaluation of $f\left(x_{i}\right)$, that is

$$
\begin{equation*}
f\left(x_{i}\right)=\int_{-\infty}^{+\infty} f(x) W\left(x-x_{i}, h\right) d x \tag{1.3}
\end{equation*}
$$

The kernel function $W\left(x-x_{i}, h\right)$ is required to have the following properties:

Unity

$$
\begin{equation*}
\int_{\Omega} W\left(x-x_{i}, h\right) d x=1 \tag{1.4a}
\end{equation*}
$$

Compact support

$$
\begin{cases}W\left(x-x_{i}, h\right) \neq 0 & \left|x-x_{i}\right|<h  \tag{1.4b}\\ W\left(x-x_{i}, h\right)=0 & \left|x-x_{i}\right| \geq h\end{cases}
$$

Dirac Delta property
$\lim _{h \rightarrow 0} W\left(x-x_{i}, h\right)=\delta\left(x-x_{i}\right)$
Positivity

$$
\begin{equation*}
W\left(x-x_{i}, h\right)>0 \tag{1.4c}
\end{equation*}
$$

From here after, we refer to $W\left(x-x_{i}, h\right)$ with $W_{i}(x)$, and the dependency on $h$ will be omitted.

The property ( 1.4 d ) is not necessary for a correct approximation, but it is introduced to avoid unphysical results such as negative density or energy.

The derivative evaluation is obtained through kernel evaluation of $f^{\prime}(x)$ and following
integration by parts:

$$
\begin{align*}
f^{\prime}\left(x_{i}\right) & =\int_{-\infty}^{+\infty} f^{\prime}(x) W_{i}(x) d x \\
& =\left[f(x) W_{i}(x)\right]_{-\infty}^{+\infty}-\int_{-\infty}^{+\infty} f(x) W_{i}^{\prime}(x) d x \tag{1.5}
\end{align*}
$$

Here, the ", " represents the derivative with respect to the independent variable $x$. Far from the boundary, that is, when $\Omega \cap \Omega_{i}=\Omega_{i}$, we assume that the smoothing function is completely developed in the domain, and thus the derivative approximation is

$$
\begin{equation*}
f^{\prime}\left(x_{i}\right)=-\int_{-\infty}^{+\infty} f(x) W_{i}^{\prime}(x) d x \tag{1.6}
\end{equation*}
$$

The higher order derivatives approximation is obtained by iterating the procedure shown in (1.6). The general formula is

$$
\begin{equation*}
f^{(n)}\left(x_{i}\right)=(-1)^{n} \int_{-\infty}^{+\infty} f(x) W_{i}^{(n)}(x) d x \tag{1.7}
\end{equation*}
$$

Equation (1.6) and its generalization (1.7) for higher order derivatives is exact far from the boundary, that is, where the first term of the right-hand side of (1.5) vanishes because of $(1.4 \mathrm{~b})$. On the other hand, when the intersection between the general domain and the smoothing length of a particle is non empty, the first term of the right-hand side of (1.5) does not vanish and the formula (1.6) does not converge. This is the most important limit of SPH approximation, that is corrected through some expedients in numerical simulations.


Figure 1.1: 1D domain: particle discretization and subdivision of the domain

Equations (1.3), (1.5) and (1.7) are still integral expression, that need a further discretization step: therefore, integrals are replaced by summations: the domain is partitioned into some little subdomains, so that there is a univocal correspondence between each particle and its subdomain (see figure 1.1). Therefore, the discrete form of (1.3) is

$$
\begin{equation*}
\int_{-\infty}^{+\infty} f(x) W_{i}(x) d x \simeq \sum_{j} f\left(x_{j}\right) W_{i}\left(x_{j}\right) \Delta x_{j} \tag{1.8}
\end{equation*}
$$

and, for higher-order derivatives

$$
\begin{equation*}
f^{(n)}\left(x_{i}\right) \simeq(-1)^{n} \sum_{j} f\left(x_{j}\right) W_{i}^{(n)}\left(x_{j}\right) \Delta x_{j} \tag{1.9}
\end{equation*}
$$

where $\Delta x_{j}$ is the reference subdomain of the particle at $x_{j}$.
We can substitute $\Delta x_{j}$ with $\frac{m_{j}}{\rho_{j}}$, where $m_{j}$ is the reference mass of the particle $j$ and $\rho_{j}$ is its density. Equation (1.9) can be rewritten in the form

$$
\begin{equation*}
f^{(n)}\left(x_{i}\right) \simeq(-1)^{n} \sum_{j} f\left(x_{j}\right) W_{i}^{(n)}\left(x_{j}\right) \frac{m_{j}}{\rho_{j}} \tag{1.10}
\end{equation*}
$$

where it is more evident the peculiarity of this method of dealing with particles.

### 1.2 Alternative approaches to SPH

In consideration of the deficiencies of the original SPH technique, especially at the boundary, many authors have introduced corrections in the most recent years. In the following paragraph we review some of the most significant numerical approximation techniques introduced starting from the initial SPH idea.

### 1.2.1 Reproducing Kernel Particle Method

One of the most common methods developed from the SPH is the Reproducing Kernel Particle Method, introduced by Liu et al. (1995a,b). In these works the authors introduce the possibility of different smoothing functions (that in these works are called window functions) for each particle in the domain, to restore the consistency of the method also close to the domain (property (1.4a)).

The method is obtained through the projection of the Taylor series expansion until the second order

$$
\begin{equation*}
f(x)=f\left(x_{i}\right)+f^{\prime}\left(x_{i}\right)\left(x-x_{i}\right)+\frac{1}{2} f^{\prime \prime}\left(x_{i}\right)\left(x-x_{i}\right)^{2}+o\left(\left|x-x_{i}\right|^{3}\right) \tag{1.11}
\end{equation*}
$$

onto a generic window function $K_{i}(x)$, that is

$$
\begin{array}{r}
\int_{\Omega} K_{i}(x) f(x) d x=f\left(x_{i}\right) \int_{\Omega} K_{i}(x) d x+f^{\prime}\left(x_{i}\right) \int_{\Omega} K_{i}(x)\left(x-x_{i}\right) d x+ \\
f^{\prime \prime}\left(x_{i}\right) \int_{\Omega} \frac{1}{2} K_{i}(x)\left(x-x_{i}\right)^{2} d x \tag{1.12}
\end{array}
$$

Then the exact reproduction of a generic function $f(x)$ can be imposed, that is

$$
\begin{equation*}
\int_{\Omega} K_{i}(x) d x=1 \tag{1.13}
\end{equation*}
$$

In this way the zero-th convergence of the method is estabilished (that is, the method
exactly reproduces constant functions). In particular, if we choose an even function, that is possible far from the boundary, also the term involving the first derivative vanishes and the method becomes second order accurate; close to the boundary, on the contrary, the method is first order accurate.

The novelty introduced by the authors is that the window function $K_{i}(x)$ can be seen as the product of two functions, $C_{i}(x)$ and $W(x)$, the second being the typical SPH kernel function, and the first being a corrective function, typically a polynomial, that is different particle by particle, which is asked to restore the required consistency conditions at particles close to the boundary. Typical condition that can be imposed to the correction function are that the high-order momenta vanish.

$$
\begin{align*}
\int_{\Omega} K_{i}(x)\left(x-x_{i}\right) d x & =0  \tag{1.14}\\
\int_{\Omega} \frac{1}{2} K_{i}(x)\left(x-x_{i}\right)^{2} d x & =0 \tag{1.15}
\end{align*}
$$

The possibility of enforcing these conditions depends on the order of the polynomial $C_{i}(x)$. After performing this kernel estimates, the evaluation of derivatives is performed according to Equations (1.6) and (1.7).

### 1.2.2 Corrective Smoothed Particle Method

The Corrective Smoothed Particle Method (Chen et al., 1999a,b) is another method developed to correct the SPH approximation techniques at the boundary. The authors do not use any correction function, but use a different kernel evaluation. Starting from the Taylor series expansion of a function $f(x)$ up to the zero-th order

$$
\begin{equation*}
f(x)=f\left(x_{i}\right) \tag{1.16}
\end{equation*}
$$

and projecting it onto a kernel function $W_{i}(x)$, we obtain the following kernel estimation

$$
\begin{equation*}
f\left(x_{i}\right)=\frac{\int_{\Omega} f(x) W_{i}(x) d x}{\int_{\Omega} W_{i}(x) d x} \tag{1.17}
\end{equation*}
$$

Equation (1.17) is equivalent to (1.3) when (1.4a) is respected; moreover, if (1.4a) is not respected, Equation (1.17) still holds. From this we conclude that for this formulation is not necessary to normalize the kernel function. Hence it is evident that this approach do not require anymore the property (1.4a), required in SPH.

Expanding up to the first order, projecting onto $W_{i}^{\prime}(x)$ and rearranging the obtained terms, we obtain that

$$
\begin{equation*}
f^{\prime}\left(x_{i}\right) \simeq \frac{\int_{\Omega}\left[f(x)-f\left(x_{i}\right)\right] W_{i}^{\prime}(x) d x}{\int_{\Omega}\left[\left(x-x_{i}\right)\right] W^{\prime}(x) d x} \tag{1.18}
\end{equation*}
$$

We notice that also in the approximation of the derivative, Equation (1.18) and Equation (1.6) differ in the denominator. Also it has to be pointed out that we project on the derivative of the kernel function and not on the kernel function itself because otherwise we would obtain
a vanishing denominator. The authors also remark that, for an higher accuracy of the function approximation, is preferable that $W_{i}^{\prime}(x)$ is an odd function, but it is not a binding property.

The approximation of the higher order derivatives can be obtained just iterating the procedure adopted in (1.17) and (1.18). It is only necessary to retain, in the Taylor series expansion, all the terms until the one of interest.

### 1.2.3 Modified Smoothed Particle Hydrodynamics

A further development of the SPH technique is the Modified Smoothed Particle Hydrodynamics (MSPH) introduced by Zhang and Batra (2004). In this case function and derivatives are not approximated in different steps, but simultaneously, through the inversion of a linear system at each particle.

$$
\left(\begin{array}{ll}
A_{11}^{i} & A_{12}^{i}  \tag{1.19}\\
A_{21}^{i} & A_{22}^{i}
\end{array}\right)\binom{f\left(x_{i}\right)}{f^{\prime}\left(x_{i}\right)}=\binom{\int_{\Omega} f(x) W_{i}(x) d x}{\int_{\Omega} f(x) W_{i}^{\prime}(x) d x}
$$

where

$$
\begin{array}{ll}
A_{11}^{i}=\int_{\Omega} W_{i}(x) d x & A_{12}^{i}=\int_{\Omega}\left(x-x_{i}\right) W_{i}(x) d x \\
A_{21}^{i}=\int_{\Omega} W_{i}^{\prime}(x) d x & A_{22}^{i}=\int_{\Omega}\left(x-x_{i}\right) W_{i}^{\prime}(x) d x \tag{1.21}
\end{array}
$$

The authors remark that the minimum number of particles to be included in the smoothing length is, in a 1 D representation, is 3 , in order to prevent the matrix $\mathbf{A}$ to be singular.

A comparison can be made between the CSPM and MSPH in terms of computational cost and accuracy. The Corrective Smoothed Particle Method is evidently more efficient in terms of computational costs, since in the MSPH there is a matrix to invert at each particle; on the contrary in the CSPM the error on higher order derivatives can be affected, through error prpagation, from the error generated in the approximation of functions and lower-order derivatives.

### 1.3 Meshless methods based on shape functions

Different meshless methods are based on shape functions: among these the most common shape functions are the Meshless Local Petrov-Galerkin shape functions, the Radial Basis Function and the Local maximum entropy shape functions. All these kinds of shape functions can be used for the development of collocation methods or for the development of methods based on Galerkin formulation.

### 1.3.1 Meshless Local Petrov-Galerkin

The Meshless Local Petrov-Galerkin (MLPG) (Atluri and Zhu, 1998) is a numerical approximation method for function and derivatives based on the use of the Moving Least Square
(MLS) shape functions. Such shape functions are written in the general form

$$
\begin{equation*}
f^{h}(\mathbf{x})=\mathbf{p}^{T}(\mathbf{x}) \mathbf{a}(\mathbf{x}) \tag{1.22}
\end{equation*}
$$

where $\mathbf{p}(\mathbf{x})$ is the vector of monomial basis in $1-\mathrm{d}, 2-\mathrm{d}$ or $3-\mathrm{d}$, and $\mathbf{a}(\mathbf{x})$ is the vector of the coefficients, which could be different point by point.

The length of the vectors $\mathbf{p}(x)$ and $\mathbf{a}(x)$ depends on the desired order of consistency (for a first order consistence, in 1d case, $\mathbf{p}(\mathbf{x})=\left[\begin{array}{ll}1 & x\end{array}\right]^{T}$, for a second order consistency $\mathbf{p}(\mathbf{x})=\left[\begin{array}{lll}1 & x & x^{2}\end{array}\right]^{T}$ and so on).

The vector $\mathbf{p}(\mathbf{x})$ is computed through the minimization of a weighted error function between the approximation $f^{h}(\mathbf{x})$ and the fictitious nodal values $\hat{f}_{h}$

$$
\begin{equation*}
J(\mathbf{x})=\sum_{i=1}^{N} w_{i}(\mathbf{x})\left[\mathbf{p}^{T}\left(\mathbf{x}_{i}\right) \mathbf{a}(\mathbf{x})-\hat{f}_{i}\right]^{2} \tag{1.23}
\end{equation*}
$$

where the $w_{i}(\mathbf{x})$ are weight functions, chosen to be non-zero in a neighborhood of the point $\mathbf{x}_{i}$, and zero elsewhere. This allows to define this method as local, in the sense that the approximation of a function and of its derivatives depends on few points close to the $\mathbf{x}_{i}$.

Hence, by minimizing this amount with respect to the coefficient vector $\mathbf{a}(\mathbf{x})$, it is possible to find the unknown coefficients $\mathbf{a}(\mathbf{x})$, then it is possible to write $f^{h}(x)$ in the form of the product of some shape functions (collected in the array $\boldsymbol{\Phi}(\mathbf{x})$ ) and the vector of the fictitious nodal values $\hat{\mathbf{f}}$. We remark that $\hat{\mathbf{f}}$ are not the nodal values, that means that they are not the precise values of the function in the point $x_{i}$, since the shape functions are not interpolatory.

$$
\begin{equation*}
f^{h}(x)=\sum_{i=1}^{N} \phi_{i}\left(x^{h}\right) \hat{f}_{i} \quad \hat{f}_{i} \neq f\left(x_{i}\right) \tag{1.24}
\end{equation*}
$$

where

$$
\begin{equation*}
\phi_{i}(\mathbf{x})=\sum_{j=1}^{m} p_{j}(\mathbf{x})\left[\mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x})\right]_{j i} \tag{1.25}
\end{equation*}
$$

and

$$
\left\{\begin{array}{l}
\mathbf{A}(\mathbf{x})=\sum_{i=1}^{n} w_{i}(\mathbf{x}) \mathbf{p}\left(\mathbf{x}_{i}\right) \mathbf{p}^{T}\left(\mathbf{x}_{i}\right)  \tag{1.26}\\
\mathbf{B}(\mathbf{x})=\left[w_{1}(\mathbf{x}) \mathbf{p}\left(\mathbf{x}_{1}\right), w_{2}(\mathbf{x}) \mathbf{p}\left(\mathbf{x}_{2}\right), \ldots, w_{n}(\mathbf{x}) \mathbf{p}\left(\mathbf{x}_{n}\right)\right]
\end{array}\right.
$$

The approximation of derivatives is simply performed by derivation of the shape functions.
The Meshless Local Petrov-Galerkin is used for the solution of elasticity problems (Atluri and Zhu, 2000), convection-diffusion problems (Lin and Atluri, 2000), Navier-Stokes equations (Lin and Atluri, 2001), thin plates (Long and Atluri, 2002).

### 1.3.2 Meshless methods based on radial basis functions

A widely used category of shape functions are the Radial Basis Functions (Buhmann, 2000), introduced by Kansa (1990b,a) and studied from a theoretical point of view by Franke and

Schaback (1998) for their use in collocation methods. Such shape functions are bell shaped functions and can be distinguished in

- Gaussian shape functions

$$
\begin{equation*}
\phi(r)=\exp \left(-r .^{2} / c^{2}\right) \tag{1.27}
\end{equation*}
$$

- Multiquadratic shape functions

$$
\begin{equation*}
\phi(r)=\left(r^{2}+c^{2}\right)^{n-3 / 2} \tag{1.28}
\end{equation*}
$$

- Logarithmic shape functions

$$
\begin{equation*}
\phi(r)=r \log (r) \tag{1.29}
\end{equation*}
$$

In all previous examples, $r=\sqrt{x^{2}+y^{2}}$ is the distance from a generic collocation point.
Radial Point Interpolation Method. In order to restore consistency in the approximation of polynomial functions, Radial Basis function have been integrated with polynomial (Wang and Liu, 2002). In the following, we report the main steps followed to construct such a kind of shape functions, based on a general distribution of points.

It is assumed that the generic function $u(\boldsymbol{x})$ is written as a linear combination of shape functions and some unknowns parameters. A generic function $u(\mathbf{x})$ is therefore approximated as

$$
\begin{equation*}
\hat{u}(\mathbf{x})=\sum_{i=1}^{N} a_{i} B_{i}^{0}(\mathbf{x})+\sum_{j=1}^{M} b_{j} P_{j}(\mathbf{x}), \quad M<N \tag{1.30}
\end{equation*}
$$

where the functions $P_{j}(\mathbf{x})$ are monomials, $a_{i}$ and $b_{j}$ are respectively the coefficients of the radial basis functions $B_{i}(\mathbf{x})$ and of the polynomial term $P_{j}^{0}(\mathbf{x})$. In particular the radial basis functions depend on the Euclidean distance $r_{i}=\sqrt{\left(x-x_{i}\right)^{2}+\left(y-y_{i}\right)^{2}}$, and for this reason is preferable to write $B_{i}(r)$ instead of $B_{i}(\mathbf{x})$.

Equation (1.30) can be rewritten in matrix form

$$
\begin{equation*}
u(\mathbf{x})=\mathbf{B}^{0}(\mathbf{x}) \mathbf{a}+\mathbf{P}^{T}(\mathbf{x}) \mathbf{b} \tag{1.31}
\end{equation*}
$$

Once written this approximation, it remains to find the unknown parameters $a_{i}$ and $b_{j}$ by imposing the interpolation conditions for the $N$ nodes of the domain, that are

$$
\begin{equation*}
\sum_{h=1}^{N} B_{i}^{0}\left(\mathbf{x}_{h}\right) a_{h}+\sum_{k=1}^{M} P_{i}\left(\mathbf{x}_{k}\right) b_{k}=u\left(\mathbf{x}_{i}\right) \tag{1.32}
\end{equation*}
$$

The other $M$ necessary conditions are

$$
\begin{equation*}
\sum_{k=1}^{M} P_{j}\left(\mathbf{x}_{k}\right) a_{k}=0 \tag{1.33}
\end{equation*}
$$

The choice of this particular set of constraints is motivated by the fact that in this way the system through which the unknown coefficients are retrieved is symmetric, due to the symmetry of the matrix $\mathbf{B}_{0}$.

The matrix form of this problem is

$$
\left(\begin{array}{cc}
\mathbf{B}^{0} & \mathbf{P}^{T}  \tag{1.34}\\
\mathbf{P} & \mathbf{0}
\end{array}\right)\binom{\mathbf{a}}{\mathbf{b}}=\binom{\hat{\mathbf{u}}}{\mathbf{0}}
$$

where the equations are particularised at each point belonging to the nodal set $X$.
Once written this problem, the coefficient vectors $\mathbf{a}$ and $\mathbf{b}$ can be found by inverting the algebraic system (1.34), that for simplicity we rewrite in the form

$$
\begin{equation*}
\mathbf{G}\binom{\mathbf{a}}{\mathbf{b}}=\binom{\hat{\mathbf{u}}}{\mathbf{0}} \tag{1.35}
\end{equation*}
$$

According to (1.30), the shape functions can be written as

$$
\phi(\mathbf{x})=\left[\begin{array}{ll}
\mathbf{B}_{0} & \mathbf{P} \tag{1.36}
\end{array}\right] \mathbf{G}^{-1}
$$

This method has some useful properties:

1. The shape functions are linearly independent for each nodal set
2. The shape functions, and above all, their derivatives, are of easy calculation;
3. The shape functions have a Dirac Delta property, that is $\phi_{i}\left(\mathbf{X}_{j}\right)=\delta_{i j}$.
4. It is always verified the zero-th and first order consistency of the approximations.
5. The shape functions have a compact support.

Delta Basis Function. A particular kind of Radial Basis Function, called Delta Basis Functions, have been introduced in Hon and Yang (2009). These functions are particularly appropriate when dealing with singularities. Here we only show an example of shape function, and present the meaning of the parameters.

$$
\begin{equation*}
\phi(x, \xi)=\sum_{n=1}^{M}\left[1-\left(\frac{n}{M+1}\right)^{2}\right]^{k} \sin \left(n \pi \frac{x+1}{2}\right) \sin \left(n \pi \frac{\xi+1}{2}\right) \tag{1.37}
\end{equation*}
$$

where $x$ is the independent variable, $\xi$ is the coordinate of the node, $M$ is an integer parameter that makes the shape function approach the Dirac Delta function, and $k$ is the so-called regularizing parameter. As it is shown in the paper under consideration, the absence of this parameter makes the function dramatically oscillate.

It should be noted that such shape functions vanish close to the boundary in the interval $\left[\begin{array}{cc}-1 & 1\end{array}\right]$. For this reason neither the collocation point nor the centers of the shape functions can be collocated in proximity of the extremes of this interval. The consequence is that all the problem involving these shape functions have to be studied in a different interval: the optimal choice, proposed in Reutskiy (2005), is to perform a linear transformation of the domain from its original coordinates to the interval $\left[\begin{array}{ll}-0.5 & 0.5\end{array}\right]$.

These shape functions work very fine in approximating almost singular functions, and also functions with high gradients, and this represents its greatest advantage, together with the
absolute simplicity of coding. The drawback of these approximant functions is that there are no properties of partition of unity or polynomial consistency.

### 1.3.3 The local max-ent approach

A different kind of shape functions proposed in recent years is the set of Local Maximum Entropy (LME) shape functions, introduced by Arroyo and Ortiz (2006, 2007), and developed in Cyron et al. (2009); Rosolen et al. (2012). The basic idea is to propose shape functions that satisfy both the principles of locality of the shape functions and of maximum entropy of the numerical scheme, meaning that the shape functions have to be the most unbiased possible, in the sense that their only have to be based on the positions of nodes, without any other external choice.

Let us consider a function $u(\mathbf{x})$. We can consider an approximation $u^{h}(\mathbf{x})$ in the form

$$
\begin{equation*}
u^{h}(\mathbf{x})=\sum_{a=1}^{N} \phi_{a}(\mathbf{x}) u_{a} \tag{1.38}
\end{equation*}
$$

where the $\phi_{a}(\mathbf{x})$ are the shape functions.
The LME shape functions are asked to respect zero-th and first order consistency conditions, that is constants and linear functions on a given nodset $\mathbf{X}$ have to be exactly reproduced:

$$
\begin{equation*}
\sum_{a=1}^{N} \phi_{a}(\mathbf{x})=1 \quad, \quad \sum_{a=1}^{N} \phi_{a}(\mathbf{x}) \mathbf{x}_{a}=\mathbf{x} \tag{1.39}
\end{equation*}
$$

In particular, the first condition of (1.39), in addition to the requirement that the shape functions have to be positive, allows to consider these approximations as a probability distribution at each point $x_{i}$.

A natural concept strictly connected to the probability is the entropy of a distribution, that is a canonical measure of the uncertainty of a distribution. To give a quick example of what the entropy represents, let us consider two distributions of probability of occurrence of two events $A_{1}$ and $A_{2}$. The first distribution is [0.01 0.99], the second is [0.50.5]. It is obvious that in the first case are quite sure about the occurrence of the event $A_{2}$, while in the second case we are in a situation of uncertainty. The quantity that measures this difference in the level of information in this cases is the entropy, that is higher for the distribution $A_{2}$.

In the case of unbiased approximations, we require that the shape functions are free of any a priori information, that is, the entropy associated to shape functions has to be the maximum, according to the Shannon theorem (Shannon, 2001). The eventual a priori choice of a polynomial shape functions of any order, or a Gaussian shape function or whatever other shape function represents a specific choice of the approximation that in this method the authors want to avoid.

The entropy of a shape function distribution, from a mathematical point of view, is defined as

$$
\begin{equation*}
H(\phi(\mathbf{x}))=-\sum_{a=1}^{N} \phi_{a}(\mathbf{x}) \log \phi_{a}(\mathbf{x}) \tag{1.40}
\end{equation*}
$$

The shape functions come from the minimization of the functional $H(\boldsymbol{\phi}(\mathbf{x}))$ with respect to the functions $\phi(\mathbf{x})$. Such problem is subject to the constraints (1.39). In their work, the authors show that a unique solution exists if and only if the data set $\mathbf{X}$ belongs to a convex hull.

Unfortunately this set of functions gives a non satisfactory result. In fact, due to the theoretical meaning of entropy, such function are as constant as possible, considering the constraints. This property returns bad approximation of non smooth functions, making this set of shape functions uninteresting.

A further step toward a best approximation is the introduction also of the attribute of locality of the shape function; the width of the approximation is defined as

$$
\begin{equation*}
w\left[\phi_{a}(\mathbf{x})\right]=\int_{\Omega} \sum_{a=1}^{N} \phi_{a}(\mathbf{x})\left|\mathbf{x}-\mathbf{x}_{a}\right|^{2} d \mathbf{x} \tag{1.41}
\end{equation*}
$$

In order to satisfy the requisite of locality of the shape functions, we also require the functional (1.41) to be minimum. We exploit the fact that in the functional itself there are no derivatives, so we minimize it pointwise. The problem so is to minimize the functional

$$
\begin{equation*}
U(\mathbf{x}, \phi(\mathbf{x}))=\sum_{a=1}^{N} \phi_{a}(\mathbf{x})\left|\mathbf{x}-\mathbf{x}_{a}\right|^{2} \tag{1.42}
\end{equation*}
$$

with respect to the same constraints (1.39). Such a problem gives back pointwise defined shape functions.

The best solution for this problem is then the search for a set of shape functions that have both the requisites of locality and impartiality. The idea is to combine the two functionals $H(\boldsymbol{\phi}(\mathbf{x}))$ and $U(\mathbf{x}, \phi(\mathbf{x}))$ through the functional

$$
\begin{equation*}
f_{\beta}(\mathbf{x}, \phi(\mathbf{x}))=\beta U(\mathbf{x}, \phi(\mathbf{x}))-H(\phi(\mathbf{x})) \tag{1.43}
\end{equation*}
$$

subject to the usual constraints (1.39). The parameter $\beta$ belongs to the interval $[0, \infty)$ and defines how much the functional is more similar to a max-ent functional $(\beta \rightarrow 0)$ or to a locality functional $(\beta \rightarrow \infty)$.

The final shape functions are then, for internal points

$$
\begin{equation*}
\phi_{\beta a}(\mathbf{x})=\frac{1}{Z\left(\mathbf{x}, \lambda^{*}(\mathbf{x})\right)} \exp \left[-\beta\left|\mathbf{x}-\mathbf{x}_{a}\right|^{2}+\lambda^{*} \cdot\left(\mathbf{x}-\mathbf{x}_{a}\right)\right] \tag{1.44}
\end{equation*}
$$

where

$$
\begin{equation*}
Z(\mathbf{x}, \lambda(\mathbf{x}))=\sum_{a=1}^{N} \exp \left[-\beta\left|\mathbf{x}-\mathbf{x}_{a}\right|^{2}+\lambda \cdot\left(\mathbf{x}-\mathbf{x}_{a}\right)\right] \tag{1.45}
\end{equation*}
$$

and

$$
\begin{equation*}
\lambda^{*}(\mathbf{x})=\arg \min _{\lambda} \log Z(\mathbf{x}, \lambda(\mathbf{x})) \tag{1.46}
\end{equation*}
$$

For the points on the boundary, the shape function are found in a similar way of (1.44),
but the summation are performed only on the particles that belong to the same face of the boundary, and not on the interior points. This is a non negligible property, because in this way it is possible to satisfy in a simple way the essential boundary conditions, that in most of the methods described earlier represent an open problem. This property is the so called weak Kronecker Delta property.

The minimization over $\lambda$ of the functional $\log Z(\mathrm{x}, \lambda(\mathrm{x}))$, is non difficult to perform, since there are no derivatives involved. For this reason the calculation can be done pointwise, and the problem reduces to a minimization over a variable $\lambda$ of a vectorial function $\log Z(\lambda)$.

Second-order max-ent. The Second-order Maximum Entropy functions have been introduced with the aim of improving the consistency order of the Max-Ent shape function, and also to remove the still existing parameter that has to be imposed by the external of the method, the parameter $\beta$, that states the grade of locality of the approximants. In this case, the generic shape functions has to obey also the condition

$$
\begin{equation*}
\sum_{a=1}^{N} \phi_{a} x_{a}^{2}=x^{2} \tag{1.47}
\end{equation*}
$$

that can be also rewritten, recalling (1.39), in the form

$$
\begin{equation*}
\sum_{a=1}^{N} \phi_{a}\left(x_{a}-x\right)^{2}=0 \tag{1.48}
\end{equation*}
$$

In these equations, and in the following, the dependence of $\phi_{a}$ on $\mathbf{x}$ will be omitted for simplicity. Recalling the interpretation of the $\phi_{a}$ as a probability distribution, the first term of Equation (1.48) can be seen as the variance of the distribution. It is obvious that it cannot be zero, otherwise it would mean that all the $\phi_{a}$ are zero except for the case $x=x_{a}$.

The solution has been proposed in (Cyron et al., 2009). Instead of condition (1.48), the authors propose

$$
\begin{equation*}
\sum_{a=1}^{N} \phi_{a}\left(x_{a}-x\right)^{2}=g(\mathbf{x}) \tag{1.49}
\end{equation*}
$$

where $g(\mathbf{x})$, called gap function, has to be chosen in a suitable way. Obviously such a correction relaxes the request of second order reproducibility of the method, but it will be clear that this relaxation only works on the boundary, leaving the second-order consistency in the interior of the domain.

In the following we describe how to choose the gap function in 1d. The procedure is easily generalizable in higher-dimensions.

The first remark is that the gap functions have to respect the boundary conditions (invoking the weak Kronecker Delta property), and so they have to vanish at the boundaries. The form of the gap function could be

$$
\begin{equation*}
g(x)=\sum_{a=1}^{N} \phi_{a, 1} w_{a} \tag{1.50}
\end{equation*}
$$

where the $\phi_{a, 1}$ are the first order Local Max-Ent shape functions and the $w_{a}$ are weights. This
choice does not hurt with the request of smoothness of the approximants, since the firs order Local Max-Ent shape function are $C^{\infty}$.

An heuristic choice of the gap function, that gives excellent results (according to the authors) is a constant function $g(x)$, smoothly ramping to zero in proximity of the boundary. This function has the form

$$
g(\xi)=\left\{\begin{array}{lc}
\frac{h_{\max }^{2}}{4} & \xi_{2}<\xi<\xi_{N-1}  \tag{1.51}\\
-\xi\left(\xi-h_{1,2}\right) & \xi_{1}<\xi<\xi_{2}
\end{array}\right.
$$

$\xi$ being a local coordinate originating on the boundary, $h_{\max }$ being the maximum distance between two particles in the domain, and $h_{1,2}$ being the distance between the boundary particle and its nearest one.

In the two and three-dimension case many other possibility are proposed for the determination of the most effective gap functions, and we do not go into details for brevity. We refer to the original works for further details.

Once defined the gap function $g(x)$, it finally remains to determine the shape functions. The request is to maximize the functional

$$
\begin{equation*}
H(\phi, \mathbf{x})=-\sum_{a=1}^{N} \phi_{a} \log \phi_{a} \tag{1.52}
\end{equation*}
$$

under the constraints (1.39) and (1.49). The solution is given by

$$
\begin{equation*}
\phi_{a}=\frac{1}{Z\left(\mathbf{x}, \lambda^{*}, \mu^{*}\right)} \exp \left[\lambda^{*}\left(x-x_{a}\right)+\mu^{*}\left[\left(x-x_{a}\right)^{2}-g(x)\right]\right], \tag{1.53}
\end{equation*}
$$

where, as earlier,

$$
\begin{equation*}
Z(x, \lambda, \mu)=\sum_{a=1}^{N} \exp \left[\lambda\left(x-x_{a}\right)+\mu\left[\left(x-x_{a}\right)^{2}-g(x)\right]\right] \tag{1.54}
\end{equation*}
$$

is the partition function, and

$$
\begin{equation*}
\left[\lambda^{*}, \mu^{*}\right]=\arg \min _{\lambda, \mu} \log Z(\lambda, \mu) \tag{1.55}
\end{equation*}
$$

Once again, the minimization can be performed pointwise, and so the functional $Z$ reduces to a function of the only unknowns $\lambda$ and $\mu$. We remark that, as stated earlier, these shape functions, differently from the first order Local Max-Ent, are naturally local, because they vanish far from $x_{a}$, and there is no need of a parameter such as $\beta$ introduced in (1.43).

A great difficulty for this method is the computation of the spatial derivatives of the shape functions: in fact, recalling (1.53), they involve the functions $Z^{*}(x), \lambda^{*}(x)$ and $\mu^{*}(x)$, that we only know pointwise.

Let's consider the gradient of the partition function (1.54).

$$
\mathbf{r}(x, \lambda, \mu)^{T}=\left[\begin{array}{ll}
\partial_{\lambda} & \partial_{\mu} \tag{1.56}
\end{array}\right] Z(x, \lambda, \mu)
$$

Recalling (1.55), we know that for $\lambda=\lambda^{*}$ and $\mu=\mu^{*}$, the functional $Z$ has a minimum $Z^{*}$ for every $x$, and so its gradient $\mathbf{r}^{*}$ is zero for each $x$.

Now we calculate the derivative of the gradient with respect to $x$. It also has to vanish for each $x$.

$$
\frac{d \mathbf{r}^{*}}{d x}=\frac{\partial \mathbf{r}^{*}}{\partial x}+\mathbf{H}^{*} \frac{d}{d x}\left[\begin{array}{c}
\lambda^{*}  \tag{1.57}\\
\mu^{*}
\end{array}\right]=0
$$

where

$$
\mathbf{H}=\left[\begin{array}{ll}
Z_{\lambda \lambda} & Z_{\lambda \mu}  \tag{1.58}\\
Z_{\lambda \mu} & Z_{\mu \mu}
\end{array}\right]
$$

is the Hessian matrix of the function $Z$ particularized at $\lambda^{*}$ and $\mu^{*}$. Now the derivatives can easily be calculated by inverting (1.57), where all the other terms are easy to calculate. Moreover, if the minimization (1.55) is performed through the method of Newton-Raphson (which is possible since the KKT conditions are respected). The Hessian is already available from the minimization procedure of (1.54).

SME - Isogeometric analysus. One of the most recent development of the Second order maximum entropy approach is the fusion with a recent successfully method based on Isogeometric Analysis (IGA). In Rosolen and Arroyo (2013) the authors start from the consideration that the so called weak Kronecker Delta property (that is the characteristic of the SME shape functions in the interior of the domain of vanishing on the boundary) holds only in a convex-hull. Conversely, the representation of the domain using the IGA, which uses the same procedure of the CAD systems for the definition of the geometry, is highly more precise than with the Second Order approach, and moreover the shape function preserve the weak Kronecker Delta property also when the collocation points do not belong to a convex hull.

Moreover, the distribution of collocation points in the IGA is by far more rigid than in the case of the maximum entropy approach. The method developed in Rosolen and Arroyo (2013) considers the best characteristics of this two approaches. On the boundary, the isogeometric shape functions are considered. In the interior, the shape function are the max-ent ones. In a strip near the boundary, where the shape functions of the boundary have not vanished at yet, the shape function are a mixture of the IGA shape functions and the max-ent ones. These are calculated always imposing the maximum entropy criterion, but also considering the contribution of the IGA shape functions.

Maximum Entropy shape functions - final considerations. The method proposed in the present sections presents many interesting characteristics, above all the one of being totally independent of external choices, and of being sufficiently robust with unstructured point distribution. Nevertheless here we remark that the robustness of the method with respect ot unstructured distributions is valid only for the first-order LME, while in second-order distributions a method for determining the gap function, and thus, a correct approximation of shape functions, has not been found. Moreover the methods looks not to be suitable for Lagrangian dynamics, since the shape function determination is a computationally expensive algorithm (due to the non-linear maximization of the entropy), that should be repeated at each particle redistribution, making the method particularly ineffective for fast dynamics.

### 1.4 Meshless methods based on the Taylor series

A different typology of meshless method is based on the Taylor series expansion: among these here we briefly review the Generalized Finite Difference Method. Another method based on the Taylor series expansion is the Least Square based Finite Difference Method (Ding et al., $2004 \mathrm{a}, \mathrm{b})$, for which we do not go into details, but we refer the reader the original works.

### 1.4.1 Generalized Finite Difference Method

The Generalized Finite Difference Method (GFDM), introduced by Benito et al. (2001), moves from a different starting point with respect to the methods discussed until now. The approximation schemes of $f^{\prime}\left(x_{i}\right)$ and $f^{\prime \prime}\left(x_{i}\right)$ are achieved by the minimization of a weighted error between the Taylor series expansion of the function $f(x)$ about a point $x_{i}$ and the value of $f\left(x_{i}\right)$ itself; the authors consider the influence region of a particle as the set of a certain number of surrounding points, selected with the so-called star criterion. The authors apply their method to hyperbolic and parabolic partial differential equations (Benito et al., 2007), to the advection-diffusion equation (Ureña et al., 2011), as well as to third- and fourth-order partial differential equations (Ureña et al., 2012; Gavete et al., 2012).

Here we report the basic idea of the Generalized Finite Difference Method (GFDM) in the multidimensional case. The starting point is the Taylor series expansion of a function $u(\boldsymbol{x})$ about a point $\boldsymbol{x}_{i}$, and its evaluation at a point $\boldsymbol{x}_{j}$. The truncation error is computed as $e_{i j}=u\left(\boldsymbol{x}_{j}\right)-u_{j}$, where $u\left(\boldsymbol{x}_{j}\right)$ is the exact value of the function in $\boldsymbol{x}_{j}$ and $u_{j}$ is its approximation. Then, a set of points $\boldsymbol{x}_{j}$ surrounding $\boldsymbol{x}_{i}$ is considered, and the total squared weighted error $E_{i}=\sum_{j}\left[e_{i j} T^{i j}\right]^{2}$ is computed, being $T^{i j}=T\left(\boldsymbol{x}_{j}-\boldsymbol{x}_{i}\right)$ a weight function, that is:

$$
\begin{align*}
E_{i} & =\sum_{j}\left\{\left[u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)+D_{x} u\left(\boldsymbol{x}_{i}\right)\left(x_{j}-x_{i}\right)+D_{y} u\left(\boldsymbol{x}_{i}\right)\left(y_{j}-y_{i}\right)+\right.\right. \\
& +D_{z} u\left(\boldsymbol{x}_{i}\right)\left(z_{j}-z_{i}\right)+\frac{1}{2} D_{x x}^{2} u\left(\boldsymbol{x}_{i}\right)\left(x_{j}-x_{i}\right)^{2}+\frac{1}{2} D_{y y}^{2} u\left(\boldsymbol{x}_{i}\right)\left(y_{j}-y_{i}\right)^{2}+  \tag{1.59}\\
& +\frac{1}{2} D_{z z}^{2} u\left(\boldsymbol{x}_{i}\right)\left(z_{j}-z_{i}\right)^{2}+D_{x y}^{2} u\left(\boldsymbol{x}_{i}\right)\left(x_{j}-x_{i}\right)\left(y_{j}-y_{i}\right)+ \\
& \left.\left.+D_{y z}^{2} u\left(\boldsymbol{x}_{i}\right)\left(y_{j}-y_{i}\right)\left(z_{j}-z_{i}\right)+D_{x z}^{2} u\left(\boldsymbol{x}_{i}\right)\left(x_{j}-x_{i}\right)\left(z_{j}-z_{i}\right)\right] T^{i j}\right\}^{2}
\end{align*}
$$

Finally the approximation schemes for the spatial derivatives are simply achieved by minimizing (1.59) with respect to the values of the derivatives in $\boldsymbol{x}_{i}$. After minimization, the
following linear system is obtained

$$
\boldsymbol{A}^{i}\left(\begin{array}{c}
D_{x} u\left(\boldsymbol{x}_{i}\right)  \tag{1.60}\\
D_{y} u\left(\boldsymbol{x}_{i}\right) \\
D_{z} u\left(\boldsymbol{x}_{i}\right) \\
D_{\boldsymbol{x}_{x}^{2}}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{\boldsymbol{x}_{y}^{2}}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{\boldsymbol{z}}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{x_{x}}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{y}^{2} u\left(\boldsymbol{x}_{2}\right) \\
D_{x z}^{2} u\left(\boldsymbol{x}_{i}\right)
\end{array}\right)=\left(\begin{array}{c}
\sum_{j} 2\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(x_{j}-x_{i}\right) T^{i j^{2}} \\
\sum_{j} 2\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(y_{j}-y_{i}\right) T^{i j^{2}} \\
\sum_{j} 2\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(z_{j}-z_{i}\right) T^{i j^{2}} \\
\sum_{j}\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(x_{j}-x_{i}\right)^{2} T^{i j^{2}} \\
\sum_{j}\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(y_{j}-y_{i}\right)^{2} T^{i j^{2}} \\
\sum_{j}\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(z_{j}-z_{i}\right)^{2} T^{i j^{2}} \\
\sum_{j} 2\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(x_{j}-x_{i}\right)\left(y_{j}-y_{i}\right) T^{i j^{2}} \\
\sum_{j} 2\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(y_{j}-y_{i}\right)\left(z_{j}-z_{i}\right) T^{i j}{ }^{2} \\
\sum_{j} 2\left(u\left(\boldsymbol{x}_{j}\right)-u\left(\boldsymbol{x}_{i}\right)\right)\left(x_{j}-x_{i}\right)\left(z_{j}-z_{i}\right) T^{i j^{2}}
\end{array}\right)
$$

For brevity we omit the expression of the components of the matrix $\boldsymbol{A}^{i}$ in the GFDM, that can be found in Benito et al. (2001, 2007).

## Chapter 2

## A Modified Finite Particle Method: 1D formulation

The Modified Finite Particle Method (MFPM) is a numerical technique for derivative approximations first introduced in Asprone et al. (2010) following the idea proposed in the SPH and successively developed in the Corrective Smoothed Particle Hydrodynamics (CSPM). In the present chapter we report the procedure adopted in the MFPM and underline the main differences with the previous formulations proposed in the literature for the monodimensional case.

In particular we start from the original formulation, proposed in Asprone et al. (2010) and then show a novel formulation, proposed for the monodimensional case in Asprone et al. (2014).

### 2.1 Modified Finite Particle Method - Derivative approximation

The approximation technique starts from the Taylor series expansion of a function $u(x)$ up to the second-order and centerd in a point $\mathbf{x}_{i}$

$$
\begin{equation*}
u(x)=u\left(x_{i}\right)+u^{\prime}\left(x_{i}\right)\left(x-x_{i}\right)+\frac{1}{2} u^{\prime \prime}\left(x_{i}\right)\left(x-x_{i}\right)^{2}+o\left(\left|x-x_{i}\right|^{3}\right) \tag{2.1}
\end{equation*}
$$

Then we project it on two projection functions $W_{1}\left(x-x_{i}\right)$ and $W_{2}\left(x-x_{i}\right)$, such that

$$
\begin{align*}
& \int_{\Omega}\left(u-u_{i}\right) W_{1}^{i} d x=u_{i}^{\prime} \int_{\Omega}\left(x-x_{i}\right) W_{1}^{i} d x+u_{i}^{\prime \prime} \int_{\Omega} \frac{1}{2}\left(x-x_{i}\right)^{2} W_{1}^{i} d x+e_{1}^{\prime}  \tag{2.2}\\
& \int_{\Omega}\left(u-u_{i}\right) W_{2}^{i} d x=u_{i}^{\prime} \int_{\Omega}\left(x-x_{i}\right) W_{2}^{i} d x+u_{i}^{\prime \prime} \int_{\Omega} \frac{1}{2}\left(x-x_{i}\right)^{2} W_{2}^{i} d x+e_{2}^{\prime}
\end{align*}
$$

Hereafter, for simplicity of notation, we denote $W_{\alpha}^{i}=W_{\alpha}\left(x-x_{i}\right)$ and $u\left(x_{i}\right)=u_{i} . e_{1}^{\prime}$ and $e_{2}^{\prime}$ are the projections of the truncation error. We note that the method is similar to the Modified Smoothed Particle Hydrodynamics (see Chapter 1), with the novelty that in this
case the function value $u_{i}$ is not considered an unknown for the approximation of derivatives.
We divide the whole domain in a finite number of subdomains, whose measure is $\Delta x_{j}$, each one referring to the particle in $x_{j}$. Then we discretize the integrals of (2.2), and we obtain a set of two equations that we may write in matrix form as

$$
\left[\begin{array}{cc}
A_{11}^{i} & A_{12}^{i}  \tag{2.3}\\
A_{21}^{i} & A_{22}^{i}
\end{array}\right]\binom{u_{i}^{\prime}}{u_{i}^{\prime \prime}}=\binom{\sum_{j}\left[u_{j}-u_{i}\right] W_{1}^{i j} \Delta x_{j}}{\sum_{j}\left[u_{j}-u_{i}\right] W_{2}^{i j} \Delta x_{j}}+\binom{e_{1}^{\prime \prime}}{e_{2}^{\prime \prime}}
$$

Equation (2.3) holds at each particle $x_{i}$. We refer to the left-hand side matrix in Equation (2.3) as $\boldsymbol{A}^{i}$, and its components are:

$$
\begin{array}{ll}
A_{11}^{i}=\sum_{j}\left(x_{j}-x_{i}\right) W_{1}^{i j} \Delta x_{j} & A_{12}^{i}=\frac{1}{2} \sum_{j}\left(x_{j}-x_{i}\right)^{2} W_{1}^{i j} \Delta x_{j}  \tag{2.4}\\
A_{21}^{i}=\sum_{j}\left(x_{j}-x_{i}\right) W_{2}^{i j} \Delta x_{j} & A_{22}^{i}=\frac{1}{2} \sum_{j}\left(x_{j}-x_{i}\right)^{2} W_{2}^{i j} \Delta x_{j}
\end{array}
$$

where $W_{\alpha}^{i j}$ stands for $W_{\alpha}\left(x_{j}-x_{i}\right)$.
We finally obtain the approximation schemes for the first and second derivatives by inverting (2.3). An accurate analysis of the error in MFPM is available in Asprone et al. (2011).

### 2.1.1 Consideration about the smoothing function

The choice of the projection functions (replacing the classical smoothing functions used in previous works) is a fundamental issue in any particle method. In classical SPH-based methods, the traditional choice of a bell-shaped function and its derivative is preferred, and many authors investigated the properties of different kernel choices, e.g., in Lucy (1977) a polynomial function was used, while in Gingold and Monaghan (1977) a Gaussian function and its derivative were adopted. The Gaussian has been since then regarded by many authors as the golden function thanks to its smoothness even for high order derivatives; on the other hand, some authors used high order B-splines (Monaghan and Lattanzio, 1985; Morris, 1996a,b), while Johnson et al. (1996) used a quadratic function. All these kernel functions have to respect the properties of unity (1.4a), compact support (1.4c), positivity, and Dirac Delta property (1.4c).

In MFPM formulation these properties are not required. In fact, the projection functions are only bases for the projection of the unknown functions and no relation with the Dirac Delta is indeed required. For this reason they may be chosen in any way such that the matrix $\boldsymbol{A}^{i}$ is non singular; the traditional choice of a even and a odd function guarantees this property, but this is not the only possibility.

In Equations (2.3) and (2.4) summations are computed over a certain number of particles, forming the stencil of the derivative approximation schemes. As an example, in order to approximate the first derivative in $x_{i}$, we may consider the particles in $x_{i-1}$, in $x_{i}$, and in $x_{i+1}$. At the boundary, we instead consider the $i$-th particle and the two closest ones.

### 2.2 Novel formulation

In this section a novel formulation is derived, by modifying the original MFPM formulation. We consider the Taylor series expansion (2.1) about the point $x_{i}$ and collect in the vector $\boldsymbol{q}^{i}$ its evaluations in a certain number of points $x_{j}$ :

$$
\begin{equation*}
u_{j}-u_{i}=u_{i}^{\prime}\left(x_{j}-x_{i}\right)+\frac{1}{2} u_{i}^{\prime \prime}\left(x_{j}-x_{i}\right)^{2}+o\left(\left|x_{j}-x_{i}\right|^{3}\right) \tag{2.5}
\end{equation*}
$$

Then we collect in another vector, namely $\boldsymbol{w}^{i}$, the evaluations of a set of projection functions $W_{\alpha}^{i j}$ in the same points $x_{j}$. We perform the scalar product $\boldsymbol{w}^{i} \cdot \boldsymbol{q}^{i}$, neglecting the truncation error; the following equation holds:

$$
\begin{equation*}
\sum_{j}\left(u_{j}-u_{i}\right) W_{\alpha}^{i j}=u_{i}^{\prime} \sum_{j}\left(x_{j}-x_{i}\right) W_{\alpha}^{i j}+\frac{1}{2} u_{i}^{\prime \prime} \sum_{j}\left(x_{j}-x_{i}\right)^{2} W_{\alpha}^{i j} \tag{2.6}
\end{equation*}
$$

and thus, for $\alpha=1,2$, we obtain:

$$
\left(\begin{array}{ll}
A_{11}^{i} & A_{12}^{i}  \tag{2.7}\\
A_{21}^{i} & A_{22}^{i}
\end{array}\right)\binom{u_{i}^{\prime}}{u_{i}^{\prime \prime}}=\binom{\sum_{j}\left[u_{j}-u_{i}\right] W_{1}^{i j}}{\sum_{j}\left[u_{j}-u_{i}\right] W_{2}^{i j}}
$$

where

$$
\begin{array}{ll}
A_{11}^{i}=\sum_{j}\left(x_{j}-x_{i}\right) W_{1}^{i j} & A_{12}^{i}=\frac{1}{2} \sum_{j}\left(x_{j}-x_{i}\right)^{2} W_{1}^{i j}  \tag{2.8}\\
A_{21}^{i}=\sum_{j}\left(x_{j}-x_{i}\right) W_{2}^{i j} & A_{22}^{i}=\frac{1}{2} \sum_{j}\left(x_{j}-x_{i}\right)^{2} W_{2}^{i j}
\end{array}
$$

We observe that the novel formulation can be easily derived from the original one by setting $\Delta x_{j}=1$ for any $j$.

### 2.2.1 Discrete form of a 1D boundary value problem using the Modified Finite Particle Method

Let us consider a generic 1D boundary value problem in the form

$$
\left\{\begin{array}{l}
\alpha \frac{\mathrm{d}^{2} u}{\mathrm{~d} x^{2}}+\beta \frac{\mathrm{d} u}{\mathrm{~d} x}+\gamma u(x)=f(x) \quad x \in(0, L)  \tag{2.9}\\
\left.u\right|_{x=0}=\bar{u}_{1} \\
\left.\frac{\mathrm{~d} u}{d x}\right|_{x=L}=\bar{u}_{L}^{\prime}
\end{array}\right.
$$

We want to reduce it in the algebraic form

$$
\begin{equation*}
K u=f \tag{2.10}
\end{equation*}
$$

where $\boldsymbol{u}$ is the vector of the values of the unknown function $u$ at $x_{i}$. The "stiffness" matrix $\boldsymbol{K}$,
for linear problems, is the linear composition of the discrete differential operators $D^{I I}, D^{I}$, and $\boldsymbol{I}$, that are the discrete counterparts of the operators $\mathrm{d}^{2} / \mathrm{d} x^{2}, \mathrm{~d} / \mathrm{d} x$ and 1. In particular, the discrete form of (2.9) reads

$$
\left\{\begin{array}{l}
\sum_{j}\left[\alpha D_{i j}^{I I}+\beta D_{i j}^{I}+\gamma \delta_{i j}\right] u_{j}=f_{i} \quad \forall i \in\{2, . ., N-1\}  \tag{2.11}\\
u_{1}=\bar{u}_{1} \\
\sum_{j} D_{N j}^{I} u_{j}=\bar{u}_{L}^{\prime}
\end{array}\right.
$$

and

$$
\left\{\begin{array}{l}
\boldsymbol{K}=\alpha \boldsymbol{D}^{I I}+\beta \boldsymbol{D}^{I}+\boldsymbol{I} \quad \forall i \in\{2, . ., N-1\}  \tag{2.12}\\
K_{11}=1, K_{1 j}=0 \quad \text { for } j \neq 1 \\
\boldsymbol{K}_{N j}=\boldsymbol{D}_{N j}^{I}
\end{array}\right.
$$

The expressions of $\boldsymbol{D}^{I I}$ and $\boldsymbol{D}^{I}$ can be derived, for the original MFPM formulation, as

$$
\begin{equation*}
D_{i j}^{I I}=E_{21}^{i}\left[W_{1}^{i j} \Delta x_{j}-\delta_{i j} \sum_{h} W_{1}^{i h} \Delta x_{h}\right]+E_{22}^{i}\left[W_{2}^{i j} \Delta x_{j}-\delta_{i j} \sum_{h} W_{2}^{i h} \Delta x_{h}\right] \tag{2.13}
\end{equation*}
$$

and

$$
\begin{equation*}
D_{i j}^{I}=E_{11}^{i}\left[W_{1}^{i j} \Delta x_{j}-\delta_{i j} \sum_{h} W_{1}^{i h} \Delta x_{h}\right]+E_{12}^{i}\left[W_{2}^{i j} \Delta x_{j}-\delta_{i j} \sum_{h} W_{2}^{i h} \Delta x_{h}\right] \tag{2.14}
\end{equation*}
$$

where $\boldsymbol{E}^{i}=\left(\boldsymbol{A}^{i}\right)^{-1}$ and $\delta_{i j}$ is the Kronecker Delta operator, that is 1 when $i=j$ and 0 otherwise.

In the novel formulation, the discrete operators $D^{I I}$ and $D^{I}$ are equal to

$$
\begin{equation*}
D_{i j}^{I I}=E_{21}^{i}\left[W_{1}^{i j}-\delta_{i j} \sum_{h} W_{1}^{i h}\right]+E_{22}^{i}\left[W_{2}^{i j}-\delta_{i j} \sum_{h} W_{2}^{i h}\right] \tag{2.15}
\end{equation*}
$$

and

$$
\begin{equation*}
D_{i j}^{I}=E_{11}^{i}\left[W_{1}^{i j}-\delta_{i j} \sum_{h} W_{1}^{i h}\right]+E_{12}^{i}\left[W_{2}^{i j}-\delta_{i j} \sum_{h} W_{2}^{i h}\right] \tag{2.16}
\end{equation*}
$$

where, again, $\boldsymbol{E}^{i}=\left(\boldsymbol{A}^{i}\right)^{-1}$. We remark that the components of the matrix $\boldsymbol{A}^{i}$ are different in the two formulations, but here we denote them with the same name since they play an identical role in both cases.

Once the matrix $\boldsymbol{K}$ is assembled according to (2.12), the array $\boldsymbol{u}$ is obtained through the solution of the linear system (2.10).

### 2.3 Applications to 1D boundary value problems

In the following we present some numerical tests, which are a good occasion to investigate some peculiarties shown by the method in the 1D case.

The test-case is a baudary value problem in the form:

$$
\begin{equation*}
u^{\prime \prime}(x)=-f(x) \tag{2.17}
\end{equation*}
$$

with a Dirichlet condition on the left side of the domain (at $x=0$ ) and a Neumann condition on the right side, at $x=1$, both equal to zero. The function $f$ is chosen to be $f(x)=e^{x}$. The corresponding exact solution is

$$
u(x)=-e^{x}+e x+1
$$

The problem is solved using a second-order MFPM, first using a uniform distribution of particles on $x$, and then using a non uniform distribution. The non-uniform distribution has been obtained by the following equation

$$
\begin{equation*}
x_{i}^{n-u}=\left(x_{i}^{u}\right)^{\alpha} \tag{2.18}
\end{equation*}
$$

where $x_{i}^{n-}{ }^{u}$ stands for the particles position in the non uniform distribution; $x_{i}^{u}$ is the particle position in the uniform distribution, $\alpha$ is the parameter that densifies the distribution on the right side of the domain, if it is less than one; it makes collocation nodes more concentrated on the left side of the domain, if it is higher than one. It is worth noting that this formula works only if $0<x<1$, otherwise such a formula has to be preceded by a "normalization" procedure, that is the position of the particles has to undergo a geometric transformation that takes them back to the reference interval $[0,1]$. The process of transformation from a uniform distribution to a non uniform one is explained in Figures 2.1(a) and 2.1(b).

Since we have a truncation up the second order of Taylor series, we expect a second order slope of the error curve in a logarithmic diagram. The error has been computed as the 2-norm of the difference between the exact solution and the approximated one.

In figure 2.2 we show the exact and a numerical solution for the problem (2.17), with a non uniform distribution of particles.

As we can see in Figures 2.3 and 2.4, we have a second order slope both in the case of a uniform distribution and in the case a non uniform distribution. We solved the problem using an odd and a even smoothing function, to ensure the matrix $\mathbf{A}$ to be non-singular.

We also remark about the case of three particles spanned by the smoothing length. In fact, when we perform the MFPM procedure at the boundary, it happens that the related matrix A becomes singular, and this does not happen when we consider more than three particles. In fact, if we consider a smoothing function which spans five particles: close to the boundary the integrals in the matrix A become, in the discrete


Figure 2.1: Creating a controlled non uniform distribution of particles


Figure 2.2: Analytical and numerical solution for the problem (2.17)


Figure 2.3: Error diagram of a uniform distribution of particles, with a Dirichlet and a Neumann condition


Figure 2.4: Error diagram of a non uniform distribution of particles, with a Dirichlet and a Neumann condition

$$
\begin{array}{r}
A_{11}=\Delta x_{1}^{2} W_{1}^{1}\left(x_{1}-x_{0}\right)+2 \Delta x_{2}^{2} W_{2}^{1}\left(x_{2}-x_{0}\right) \\
A_{12}=\frac{1}{2}\left[\Delta x_{1}^{3} W_{1}^{1}\left(x_{2}-x_{0}\right)+4 \Delta x_{2}^{3} W_{2}^{1}\left(x_{2}-x_{0}\right)\right] \\
A_{21}=\Delta x_{1}^{2} W_{1}^{2}\left(x_{1}-x_{0}\right)+2 \Delta x_{2}^{2} W_{2}^{2}\left(x_{2}-x_{0}\right) \\
A_{22}=\frac{1}{2}\left[\Delta x_{1}^{3} W_{1}^{2}\left(x_{2}-x_{0}\right)+4 \Delta x_{2}^{3} W_{2}^{2}\left(x_{2}-x_{0}\right)\right] \tag{2.22}
\end{array}
$$

By dividing the terms of the first row by the ones of the second row of hte matrix $\mathbf{A}$ we obtain

$$
\begin{align*}
& \frac{A_{11}}{A_{21}}=\frac{W_{0}^{1}\left(x_{1}\right)+2 W_{0}^{1}\left(x_{2}\right)}{W_{0}^{2}\left(x_{1}\right)+2 W_{0}^{2}\left(x_{2}\right)}  \tag{2.23}\\
& \frac{A_{12}}{A_{22}}=\frac{W_{0}^{1}\left(x_{1}\right)+4 W_{0}^{1}\left(x_{2}\right)}{W_{0}^{2}\left(x_{1}\right)+4 W_{0}^{2}\left(x_{2}\right)} \tag{2.24}
\end{align*}
$$

In this way (2.23) is different from (2.24). But if we consider a kernel function which does not span the external particles, (that is the same of considering $W_{0}^{1}\left(x_{2}\right)=0$ and $W_{0}^{2}\left(x_{2}\right)=0$ ) we notice that (2.23) becomes equal to (2.24), and therefore the matrix $\mathbf{A}$ becomes singular. This is not a problem when we solve a Boundary Value Problem with only Dirichlet conditions, but in the case of a problem that involves derivatives in the boundary conditions, we have to invert the matrix A related to the boundary particle (in order to calculate the derivatives) and this is not possible.

### 2.3.1 Comparison with the Finite Difference Method

It can be observed that, when only three particles in the stencil are used, the second-order approximation schemes of derivatives are independent of the choice of the projection functions. In order to prove this, we recall the classical procedure to get the derivative approximation schemes used in the Finite Difference Method (LeVeque, 1955).

The approximation of a function $u$ at a point $x_{j}$ can be achieved by the Taylor expansion about $x_{i}$ up to the desired order.

A generic approximation scheme of the second derivative reads

$$
\begin{equation*}
u_{i}^{\prime \prime}=\sum_{j} \alpha_{j} u_{j} \tag{2.25}
\end{equation*}
$$

where $\alpha_{j}$ are coefficients to be determined. By combining (2.5) and (2.25), we obtain (LeVeque, 1955, see)

$$
\begin{equation*}
u_{i}^{\prime \prime}=u_{i} \sum_{j} \alpha_{j}+u_{i}^{\prime} \sum_{j} \alpha_{j}\left(x_{j}-x_{i}\right)+u_{i}^{\prime \prime} \sum_{j} \frac{1}{2} \alpha_{j}\left(x_{j}-x_{i}\right)^{2} \tag{2.26}
\end{equation*}
$$

The points $x_{i}$ are called collocation points of the approximation schemes, while the points $x_{j}$ are the points of the stencil of the approximation. In most of the schemes, they do coincide, but this is not mandatory in general.

Equation (2.26) holds only if

$$
\left\{\begin{array}{l}
\sum_{j} \alpha_{j}=0  \tag{2.27}\\
\sum_{j} \alpha_{j}\left(x_{j}-x_{i}\right)=0 \\
\sum_{j} \frac{1}{2} \alpha_{j}\left(x_{j}-x_{i}\right)^{2}=1
\end{array}\right.
$$

When the particles are three, the number of equations is equal to the number of the unknowns $\alpha_{j}$. If there are no coincident particles at the same point $x_{j}$, the system (2.27) has only one solution. Hence, the second order approximation scheme for the second derivative is unique. We remark that the same argument holds also for a second order approximation of the first derivative.

In both the original and the novel formulation of the MFPM, the final form of the derivative approximation schemes is identical to (2.25). In order to achieve a second order scheme for the first and second derivatives, any procedure has to return the same $\alpha_{j}$ of the FDM, otherwise the method would not be second-order accurate. Both original and novel MFPM satisfy this requirement.

## Chapter 3

## Multidimensional Modified Finite Particle Method

In the present chapter we extend the Modified Finite Particle Method to the multi-dimensional case. In a first moment, we will show the three-dimensional original formulation of the MFPM, as presented in Asprone et al. (2014), then we present a novel formulation, as presented in Asprone et al. (2014), and show the obtained advantages in terms of computational cost and error.

### 3.1 Original formulation

We consider the Taylor series expansion of an unknown function $u(\boldsymbol{x})$ up to the second order

$$
\begin{align*}
u(\boldsymbol{x})=u\left(\boldsymbol{x}_{i}\right) & +D_{x} u\left(\boldsymbol{x}_{i}\right)\left(x-x_{i}\right)+D_{y} u\left(\boldsymbol{x}_{i}\right)\left(y-y_{i}\right)+D_{z} u\left(\boldsymbol{x}_{i}\right)\left(z-z_{i}\right)+ \\
& +\frac{1}{2} D_{x x}^{2} u\left(\boldsymbol{x}_{i}\right)\left(x-x_{i}\right)^{2}+\frac{1}{2} D_{y y}^{2} u\left(\boldsymbol{x}_{i}\right)\left(y-y_{i}\right)^{2}+\frac{1}{2} D_{z z}^{2} u\left(\boldsymbol{x}_{i}\right)\left(z-z_{i}\right)^{2}+  \tag{3.1}\\
& +D_{x y}^{2} u\left(\boldsymbol{x}_{i}\right)\left(x-x_{i}\right)\left(y-y_{i}\right)+D_{y z}^{2} u\left(\boldsymbol{x}_{i}\right)\left(y-y_{i}\right)\left(z-z_{i}\right)+ \\
& +D_{x z}^{2} u\left(\boldsymbol{x}_{i}\right)\left(x-x_{i}\right)\left(z-z_{i}\right)+o\left(\left\|\boldsymbol{x}-\boldsymbol{x}_{i}\right\|^{3}\right)
\end{align*}
$$

where $\boldsymbol{x}=\left[\begin{array}{lll}x & y & z\end{array}\right]^{T}$.

$$
\begin{align*}
& D_{x} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(x-x_{i}\right) W_{\alpha}^{i} d V+D_{y} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(y-y_{i}\right) W_{\alpha}^{i} d V+D_{z} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(z-z_{i}\right) W_{\alpha}^{i} d V+ \\
+ & \frac{1}{2} D_{x x}^{2} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(x-x_{i}\right)^{2} W_{\alpha}^{i} d V+\frac{1}{2} D_{y y}^{2} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(y-y_{i}\right)^{2} W_{\alpha}^{i} d V+ \\
+ & \frac{1}{2} D_{z z}^{2} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(z-z_{i}\right)^{2} W_{\alpha}^{i} d V+D_{x y}^{2} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(x-x_{i}\right)\left(y-y_{i}\right) W_{\alpha}^{i} d V+ \\
+ & D_{y z}^{2} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(y-y_{i}\right)\left(z-z_{i}\right) W_{\alpha}^{i} d V+D_{x z}^{2} u\left(\boldsymbol{x}_{i}\right) \int_{\Omega}\left(x-x_{i}\right)\left(z-z_{i}\right) W_{\alpha}^{i} d V= \\
= & \int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{\alpha}^{i} d V \tag{3.2}
\end{align*}
$$

which can be rewritten in matrix form as:

$$
A^{i}\left(\begin{array}{c}
D_{x} u\left(\boldsymbol{x}_{i}\right)  \tag{3.3}\\
D_{y} u\left(\boldsymbol{x}_{i}\right) \\
D_{z} u\left(\boldsymbol{x}_{i}\right) \\
D_{\boldsymbol{x}}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{y y}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{z z}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{x y}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{y z}^{2} u\left(\boldsymbol{x}_{i}\right) \\
D_{x z}^{2} u\left(\boldsymbol{x}_{i}\right)
\end{array}\right)=\left(\begin{array}{c}
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{1}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{2}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{3}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{4}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{5}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{6}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{7}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{8}^{i} d V \\
\int_{\Omega}\left(u(\boldsymbol{x})-u\left(\boldsymbol{x}_{i}\right)\right) W_{9}^{i} d V
\end{array}\right)
$$

The choice of the projection functions is performed with the only requirement that, for each particle, the matrix $\boldsymbol{A}^{i}$ is non singular. For instance, in our tests we choose

$$
\begin{array}{lll}
W_{1}^{i}=x-x_{i} & W_{4}^{i}=\left(x-x_{i}\right)^{2} & W_{7}^{i}=\left(x-x_{i}\right)\left(y-y_{i}\right) \\
W_{2}^{i}=y-y_{i} & W_{5}^{i}=\left(y-y_{i}\right)^{2} & W_{8}^{i}=\left(y-y_{i}\right)\left(z-z_{i}\right) \\
W_{3}^{i}=z-z_{i} & W_{6}^{i}=\left(z-z_{i}\right)^{2} & W_{9}^{i}=\left(x-x_{i}\right)\left(z-z_{i}\right)
\end{array}
$$

The domain $\Omega$ is then divided into finite subdomains $\Delta V_{j}$, one for each particle $\boldsymbol{x}_{j}$, e.g., according to the Voronoi tessellation procedure (see Aurenhammer (1991) for details); for each particle an influence region $\Omega_{i}$ is also defined, depending, as in SPH-based methods, on the smoothing length. In MFPM we do not define a fixed value of the smoothing length, but we prefer to set the number of particles to be included in the domain $\Omega_{i}$ for the approximation of derivatives. For all particles such that $x_{j} \notin \Omega_{i}$ we pose that $W_{\alpha}^{i}\left(\boldsymbol{x}=\boldsymbol{x}_{j}\right)=0$ for $\alpha=1, \ldots, 9$. Then, the integrals are approximated with summations, and Equation (3.3) can be rewritten as

$$
\begin{equation*}
\boldsymbol{A}^{i} \boldsymbol{D}\left(u_{i}\right)=\boldsymbol{C}^{i} \boldsymbol{u}-\boldsymbol{B} u_{i} \tag{3.4}
\end{equation*}
$$

where

$$
\begin{align*}
C^{i} & =\left[\begin{array}{lllllll}
W^{i 1} & \mid & W^{i 2} & \mid & \ldots & \mid & W^{i N}
\end{array}\right]  \tag{3.5}\\
B^{i} & =\sum_{j} W^{i j} \tag{3.6}
\end{align*}
$$

and

$$
W^{i j}=\left[\begin{array}{llllll}
W_{1}^{i j} \Delta V_{j} & \mid & W_{2}^{i j} \Delta V_{j} & \mid & \ldots & \mid  \tag{3.7}\\
W_{9}^{i j} \Delta V_{j}
\end{array}\right]^{T}
$$

Then, by inverting (3.4), we obtain

$$
\begin{equation*}
\boldsymbol{D}\left(u_{i}\right)=\boldsymbol{E}^{i}\left(\boldsymbol{C}^{i} \boldsymbol{u}-\boldsymbol{B}^{\boldsymbol{i}} u_{i}\right)=\boldsymbol{E}^{i}\left(\boldsymbol{C}^{i}-\mathbb{B}^{i}\right) \boldsymbol{u} \tag{3.8}
\end{equation*}
$$

where

$$
\mathbb{B}^{i}=\left[\begin{array}{llllllllllllll}
0 & \mid & 0 & \mid & \ldots & \mid & B^{i} & \mid & \ldots & \mid & 0 & \mid & 0 \tag{3.9}
\end{array}\right]
$$

and finally

$$
\begin{equation*}
\boldsymbol{D}\left(u_{i}\right)=\mathbb{D}^{i} \boldsymbol{u} \tag{3.10}
\end{equation*}
$$

The operator that, applied to $\boldsymbol{u}$, gives back the discrete form of the generic derivative of $u(x)$ can be built by collecting the corresponding rows of $\mathbb{D}^{i}=\boldsymbol{E}^{i}\left(\boldsymbol{C}^{i}-\mathbb{B}^{i}\right), \forall i$.

In order to find the correct row of $\mathbb{D}^{i}$, it is sufficient to refer to Equation (3.3). For instance, in order to build the operator $\boldsymbol{D}_{x x}$ (that, applied to $\boldsymbol{u}$, gives back the discrete counterpart of $\partial^{2} u(x) / \partial x^{2}$ ), we select, for each $i$, the 4 -th row of $\mathbb{D}^{i}$, such that

$$
\boldsymbol{D}_{x x}=\left[\begin{array}{c}
\mathrm{D}_{4}^{1}  \tag{3.11}\\
\mathrm{D}_{4}^{2} \\
\ldots \\
\ldots \\
\mathbb{D}_{4}^{N}
\end{array}\right]
$$

where $\mathbb{D}_{\alpha}^{i}$ is the $\alpha$-th row of $\mathbb{D}^{i}$.
A 2D formulation of the method is simply achieved by considering only the derivatives in the $x$ and $y$ directions, that is, $\alpha$ can only be equal to $1,2,4,5,7$. The three-dimensional subdomains $\Delta V_{j}$ are obviously replaced by planar subdomains $\Delta A_{j}$.

### 3.2 Novel formulation

In the present section we introduce the novel formulation of the Modified Finite Particle Method (MFPM) as presented in Asprone et al. (2014), and use this method for the derivative approximation of a scalar function $u(\mathbf{x})$ defined in a three-dimensional domain $\Omega$, hence $\mathbf{x}=$ $\left[\begin{array}{lll}x & y & z\end{array}\right]^{T} \in \Omega \subset \mathbb{R}^{3}$. We discretize the domain $\Omega$ into a set of points $\mathbf{x}_{i}$ collected in the node set $X$.

For each point $\mathbf{x}_{i} \in X$, the approximation procedure considers the Taylor series expansion
of $u(\mathbf{x})$ up to the second order, centered in $\mathbf{x}_{i}$ :

$$
\begin{align*}
u(\mathbf{x})-u\left(\mathbf{x}_{i}\right) & =D_{x} u\left(\mathbf{x}_{i}\right)\left(x-x_{i}\right)+D_{y} u\left(\mathbf{x}_{i}\right)\left(y-y_{i}\right)+D_{z} u\left(\mathbf{x}_{i}\right)\left(z-z_{i}\right)+ \\
& +\frac{1}{2} D_{x x}^{2} u\left(\mathbf{x}_{i}\right)\left(x-x_{i}\right)^{2}+\frac{1}{2} D_{y y}^{2} u\left(\mathbf{x}_{i}\right)\left(y-y_{i}\right)^{2}+\frac{1}{2} D_{z z}^{2} u\left(\mathbf{x}_{i}\right)\left(z-z_{i}\right)^{2}+  \tag{3.12}\\
& +D_{x y}^{2} u\left(\mathbf{x}_{i}\right)\left(x-x_{i}\right)\left(y-y_{i}\right)+D_{y z}^{2} u\left(\mathbf{x}_{i}\right)\left(y-y_{i}\right)\left(z-z_{i}\right)+ \\
& +D_{x z}^{2} u\left(\mathbf{x}_{i}\right)\left(x-x_{i}\right)\left(z-z_{i}\right)
\end{align*}
$$

Then, for each $\mathbf{x}_{i}$ we select a node subset $X_{i} \subset X$, which serves as support for the derivative approximation in $\mathbf{x}_{i}$. Conceptually $X_{i}$ could coincide with the whole set of nodes $X$, but the choice of a limited number $N_{i}$ of "supporting nodes" has a beneficial effect on the final computational cost of the method.

Equation (3.12) is then evaluated in the points $\mathbf{x}_{j} \in X_{i}$, yielding

$$
\begin{align*}
u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right) & =D_{x} u\left(\mathbf{x}_{i}\right)\left(x_{j}-x_{i}\right)+D_{y} u\left(\mathbf{x}_{i}\right)\left(y_{j}-y_{i}\right)+D_{z} u\left(\mathbf{x}_{i}\right)\left(z_{j}-z_{i}\right)+ \\
& +\frac{1}{2} D_{x x}^{2} u\left(\mathbf{x}_{i}\right)\left(x_{j}-x_{i}\right)^{2}+\frac{1}{2} D_{y y}^{2} u\left(\mathbf{x}_{i}\right)\left(y_{j}-y_{i}\right)^{2}+\frac{1}{2} D_{z z}^{2} u\left(\mathbf{x}_{i}\right)\left(z_{j}-z_{i}\right)^{2}+ \\
& +D_{x y}^{2} u\left(\mathbf{x}_{i}\right)\left(x_{j}-x_{i}\right)\left(y_{j}-y_{i}\right)+D_{y z}^{2} u\left(\mathbf{x}_{i}\right)\left(y_{j}-y_{i}\right)\left(z_{j}-z_{i}\right)+ \\
& +D_{x z}^{2} u\left(\mathbf{x}_{i}\right)\left(x_{j}-x_{i}\right)\left(z_{j}-z_{i}\right) \tag{3.13}
\end{align*}
$$

It is important to highlight that at this stage we consider to know the nodal values of $u$ (i.e., $u\left(\mathbf{x}_{i}\right)$ and $u\left(\mathbf{x}_{j}\right)$ ), and, therefore, in Equation (3.13) the unknown terms are the derivative evaluations at the point $\mathbf{x}_{i}$. In order to compute such derivative values, we introduce nine arbitrary functions (referred as projection functions) $W_{\alpha}^{i}(\mathbf{x})=W_{\alpha}\left(\mathbf{x}-\mathbf{x}_{i}\right)$, with $\alpha=1, \ldots, 9$, and evaluate them at the points $\mathbf{x}_{j} \in X_{i}$.

We then multiply both sides of Equation (3.13) by the evaluations $W_{\alpha}^{i j}=W_{\alpha}\left(\mathbf{x}_{j}-\mathbf{x}_{i}\right)$ of the projection functions at the same points $\mathbf{x}_{j}$. Finally we sum all products, obtaining nine expressions of the following type

$$
\begin{align*}
& D_{x} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(x_{j}-x_{i}\right) W_{\alpha}^{i j}+D_{y} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(y_{j}-y_{i}\right) W_{\alpha}^{i j}+D_{z} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(z_{j}-z_{i}\right) W_{\alpha}^{i j}+ \\
+\quad & \frac{1}{2} D_{x x}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(x_{j}-x_{i}\right)^{2} W_{\alpha}^{i j}+\frac{1}{2} D_{y y}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(y_{j}-y_{i}\right)^{2} W_{\alpha}^{i j}+ \\
+\quad & \frac{1}{2} D_{z z}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(z_{j}-z_{i}\right)^{2} W_{\alpha}^{i j}+D_{x y}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(x_{j}-x_{i}\right)\left(y_{j}-y_{i}\right) W_{\alpha}^{i j}+ \\
+ & D_{y z}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(y_{j}-y_{i}\right)\left(z_{j}-z_{i}\right) W_{\alpha}^{i j}+D_{x z}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(x_{j}-x_{i}\right)\left(z_{j}-z_{i}\right) W_{\alpha}^{i j}= \\
= & \sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{\alpha}^{i j} \quad \alpha=1, \ldots, 9 \tag{3.14}
\end{align*}
$$

Remark. It is important, at this stage, to highlight the formal difference between the novel formulation just described, and the original formulation described in the previous section. In fact, in this case we first evaluate the Taylor Series (3.12) and the projection function, and then we project vectors, obtaining an already discretized projection. In the case of the original MFPM formulation, conversely, we perform a continous projection, obtaining an integral expression that has to be furthermore discretized, introducing an additional source of error.

Equations (3.14), repeated for $\alpha=1, \ldots, 9$, can be rearranged in matrix form as

$$
\mathbf{A}^{i}\left(\begin{array}{c}
D_{x} u\left(\mathbf{x}_{i}\right)  \tag{3.15}\\
D_{y} u\left(\mathbf{x}_{i}\right) \\
D_{z} u\left(\mathbf{x}_{i}\right) \\
D_{x}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{y,}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{z u}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{x}^{2} u u\left(\mathbf{x}_{i}\right) \\
D_{y}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{x z}^{2} u\left(\mathbf{x}_{i}\right)
\end{array}\right)=\left(\begin{array}{c}
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{1}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{2}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{3}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{4}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{5}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{6}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{7}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{8}^{i j} \\
\sum_{j}\left[u\left(\mathbf{x}_{j}\right)-u\left(\mathbf{x}_{i}\right)\right] W_{9}^{i j}
\end{array}\right)
$$

Equation (3.15) can be rearranged in a more compact form as

$$
\begin{equation*}
\mathbf{A}^{i} \mathbf{D}\left(u_{i}\right)=\mathbb{W}^{i} \mathbf{u}_{i}-\mathbf{B}^{i} u\left(\mathbf{x}_{i}\right) \tag{3.16}
\end{equation*}
$$

where $\mathrm{W}^{i}$ is a $9 \mathrm{x} N_{i}$ matrix defined as

$$
\mathbb{W}^{i}=\left[\begin{array}{llllll}
\mathbf{W}^{i 1} & \mid & \mathbf{W}^{i 2} & \mid & \ldots & \mathbf{W}^{i N_{i}} \tag{3.17}
\end{array}\right]
$$

$\mathbf{B}^{i}$ is a $9 \times 1$ vector defined as

$$
\begin{equation*}
\mathbf{B}^{i}=\sum_{j=1}^{N_{i}} \mathbf{W}^{i j} \tag{3.18}
\end{equation*}
$$

In Equations (3.17) and (3.18), $\mathbf{W}^{i j}$ is a $9 \times 1$ vector defined as

$$
\mathbf{W}^{i j}=\left[\begin{array}{lllllll}
W_{1}^{i j} & \mid & W_{2}^{i j} & \mid & \ldots & W_{9}^{i j} \tag{3.19}
\end{array}\right]^{T}
$$

that collects the evaluations of the projection functions $W_{\alpha}^{i j}$ at the nodes $\mathbf{x}_{j}$ and $\mathbf{x}_{i}$, for $\alpha=1, \ldots, 9$; finally $\mathbf{u}_{i}$ is the $N_{i} \mathrm{x} 1$ vector collecting the values of $u(\mathbf{x})$ at all nodes of the subset $X_{i}$.

Equation (3.16) can be furthermore rearranged in the form

$$
\begin{equation*}
\mathbf{A}^{i} \mathbf{D}\left(u_{i}\right)=\left(\mathbb{W}^{i}-\mathbb{B}^{i}\right) \mathbf{u}_{i} \tag{3.20}
\end{equation*}
$$

where

$$
\mathbb{B}^{i}=\left[\begin{array}{lllll|l|l|l|ll}
\mathbf{0} & \mid & \mathbf{0} & \ldots & \mathbf{B}^{i} & \ldots & \mathbf{0} & \mathbf{0} \tag{3.21}
\end{array}\right]
$$

is a $9 \mathrm{x} N_{i}$ matrix, composed of zero entries, with exception of the $i$-th column.
Derivative values at the point $\mathbf{x}_{i}$ can be retrieved inverting Equation (3.20), yielding

$$
\begin{equation*}
\mathbf{D}\left(u_{i}\right)=\mathbf{E}^{i}\left(\mathbb{W}^{i}-\mathbb{B}^{i}\right) \mathbf{u}_{i} \tag{3.22}
\end{equation*}
$$

where

$$
\begin{equation*}
\mathbf{E}^{i}=\left(\mathbf{A}^{i}\right)^{-1} \tag{3.23}
\end{equation*}
$$

Finally, Equation (3.22) can be rewritten in the form

$$
\begin{equation*}
\mathbf{D}\left(u_{i}\right)=\mathbb{D}^{i} \mathbf{u}_{i} \tag{3.24}
\end{equation*}
$$

where the matrix $\mathbb{D}^{i}=\mathbf{E}^{i}\left(\mathbb{W}^{i}-\mathbb{B}^{i}\right)$ is a $9 \mathrm{x} N_{i}$ operator acting on the vector $\mathbf{u}_{i}$ and returning the evaluations of the function derivatives at $\mathbf{x}_{i}$.

However, keeping in mind that our final goal is to solve boundary value problems governed by partial differential equations, we need nine discrete differential operators that approximate derivative operations on continous functions. In particular, recalling that $N$ is the total number of nodes used for the domain discretization, we wish to build operators that act on the $N \mathrm{x} 1$ vector $\mathbf{u}$ (collecting the function evaluations at all nodes of the domain) and reconstruct the function derivative evaluations in all nodes. Such operators are constructed collecting the proper row from the operators $\mathbb{D}^{i}, \forall i$, identified with reference to Equation (3.15). For instance, in order to construct an operator $\mathbf{D}_{x x}$ acting on $\mathbf{u}$ and returning the discrete counterpart of $\partial^{2} u(\mathbf{x}) / \partial x^{2}$, we extract, for each $i$, the 4 -th row of $\mathbb{D}^{i}$, that is

$$
\mathbf{D}_{x x}=\left[\begin{array}{c}
\mathrm{D}_{4}^{1}  \tag{3.25}\\
\mathbb{D}_{4}^{2} \\
\ldots \\
\mathrm{D}_{4}^{N}
\end{array}\right]
$$

where $\mathbb{D}_{\alpha}^{i}$ is the $\alpha$-th row of $\mathbb{D}^{i}$.
A 2 D formulation of the method is simply achieved by considering only the derivatives in the $x$ and $y$ directions, that is, $\alpha$ can only be equal to $1,2,4,5,7$.

Given the analytical form of derivative approximation schemes, it is easy to approximate any linear differential operator; the most common are reported in Table 3.1

| Continuum operator | Discrete operator |
| :---: | :---: |
| $\boldsymbol{\nabla} \cdot(\bullet)$ | $\mathbf{D}_{x}(\bullet)_{1}+\mathbf{D}_{y}(\bullet)_{2}+\mathbf{D}_{z}(\bullet)_{3}$ |
| $\boldsymbol{\nabla}(\bullet)$ | $\left[\mathbf{D}_{x} \quad \mathbf{D}_{y} \quad \mathbf{D}_{z}\right]^{T}(\bullet)$ |
| $\boldsymbol{\nabla}^{2}(\bullet)$ | $\mathbf{D}_{x x}(\bullet)+\mathbf{D}_{y y}(\bullet)+\mathbf{D}_{z z}(\bullet)$ |

Table 3.1: Correspondence between some continuum differential operators and their discrete form using MFPM.

### 3.2.1 Projection functions and supporting nodes

In the following we give some selection criteria for the projection functions and for the "supporting nodes".

The projection functions $W_{\alpha}^{i}$ have to be chosen such that matrix $\mathbf{A}^{i}$ is non singular. In all the examples of the present paper, we use linear, quadratic and bilinear functions in the three independent variables, i.e.:

$$
\begin{array}{lll}
W_{1}^{i}=x-x_{i} & W_{4}^{i}=\left(x-x_{i}\right)^{2} & W_{7}^{i}=\left(x-x_{i}\right)\left(y-y_{i}\right) \\
W_{2}^{i}=y-y_{i} & W_{5}^{i}=\left(y-y_{i}\right)^{2} & W_{8}^{i}=\left(y-y_{i}\right)\left(z-z_{i}\right) \\
W_{3}^{i}=z-z_{i} & W_{6}^{i}=\left(z-z_{i}\right)^{2} & W_{9}^{i}=\left(x-x_{i}\right)\left(z-z_{i}\right)
\end{array}
$$

The dimensions of the subsets $X_{i}$, that contain the supporting nodes for each $\mathbf{x}_{i}$, is not specified a priori. The number of supporting nodes, as well as their criterion of selection, is the topic of several works in the literature regarding meshless methods. In the original version of the Smoothing Particle Hydrodynamics a smoothing length is selected, and all particles whose distance is less than the fixed smoothing length are used for the approximation. On the other hand, in the Generalized Finite Difference Method, the selection criterion includes the particles $\mathbf{x}_{i}$ and the two closest ones for each quadrant of a local reference frame, centered on the particle itself and with the axes parallel to the global axes (star criterion). In the LeastSquare Finite Difference Method, proposed by Ding et al. (2004a,b) the selection algorithm of supporting nodes considers the $N_{i}$ nodes closest to $\mathbf{x}_{i}$, and then a check is performed on the condition number of the local system to be solved (with a structure similar to (3.15)). The authors observe that when one or more supporting nodes are very close to $\mathbf{x}_{i}$, the matrix $\mathbf{A}^{i}$ may result ill-conditioned. Therefore a scaling matrix $\mathbf{S}$, based only on the reciprocal distance among particles, is introduced in order to improve the quality of $\mathbf{A}^{i}$, and thus, of the derivative approximations.

We use a selection algorithm very similar to the one proposed in the LSFDM, and composed of two steps. First, a predetermined number $N_{i}$ of particles is included in the set of supporting nodes $X_{i}$, selected only on the base of their distance from the particle $\mathbf{x}_{i}$. Then, a diagonal scaling matrix $\mathbf{S}$ is introduced, whose expression is

$$
\begin{equation*}
\mathbf{S}=\operatorname{diag}\left[1 / R, 1 / R, 1 / R, 1 / R^{2}, 1 / R^{2}, 1 / R^{2}, 1 / R^{2}, 1 / R^{2}, 1 / R^{2}\right] \tag{3.26}
\end{equation*}
$$

where $R=\max \left\|\mathbf{x}_{j}-\mathbf{x}_{i}\right\|, \mathbf{x}_{j} \in X_{i}$. Equation (3.15) is then rewritten introducing the matrix $\mathbf{S}$ both in the left and right term. The condition number of the resulting matrix $\overline{\mathbf{A}}^{i}=\mathbf{S} \mathbf{A}^{i}$ is computed; if such condition number is greater than a fixed threshold value $C_{\text {max }}$, the number of supporting nodes is increased and the procedure is repeated, until a satisfactory condition number is reached.

Accordingly, Equation (3.4) can be rewritten in the form

$$
\begin{equation*}
\overline{\mathbf{A}}^{i} \mathbf{D}\left(u_{i}\right)=\mathbf{S A}^{i} \mathbf{D}\left(u_{i}\right)=\mathbf{S A}^{i}\left(\mathbb{W}^{i} \mathbf{u}_{i}-\mathbf{B}^{i} u\left(\mathbf{x}_{i}\right)\right) \tag{3.27}
\end{equation*}
$$

and all equations from (3.15) to (3.23) are properly rearranged.

### 3.3 Comparison with existing methods

By comparing Equations (3.3), (3.15), and (1.60), we notice the similarity between the GFDM and both the original and novel formulations of the MFPM. In particular, it is easy to see that, with a proper choice of the projection functions in the MFPM, the GFDM is recovered. In particular, the GFDM and the novel MFPM exactly match, if the following functions are chosen for the novel MFPM:

$$
\begin{gather*}
W_{1}^{i j}=2\left(x_{j}-x_{i}\right) T^{i j^{2}} \\
W_{2}^{i j}=2\left(y_{j}-y_{i}\right) T^{i j^{2}} \\
W_{3}^{i j}=2\left(z_{j}-z_{i}\right) T^{i j^{2}} \\
W_{4}^{i j}=\left(x_{j}-x_{i}\right)^{2} T^{i j^{2}} \\
W_{5}^{i j}=\left(y_{j}-y_{i}\right)^{2} T^{i j^{2}}  \tag{3.28}\\
W_{6}^{i j}=\left(z_{j}-z_{i}\right)^{2} T^{i j^{2}} \\
W_{7}^{i j}=2\left(x_{j}-x_{i}\right)\left(y_{j}-y_{i}\right) T^{i j^{2}} \\
W_{8}^{i j}=2\left(y_{j}-y_{i}\right)\left(z_{j}-z_{i}\right) T^{i j^{2}} \\
W_{9}^{i j}=2\left(x_{j}-x_{i}\right)\left(z_{j}-z_{i}\right) T^{i j^{2}}
\end{gather*}
$$

We remark that in the GFDM all the derivative approximation schemes depend on the choice of one weight function $T_{i j}$, while in the MFPM formulations 9 projection functions have to be defined.

### 3.3.1 Considerations about the stencil of the derivatives

The stencil of a derivative approximation is the set of particles which contributes to the approximation in a point $\mathbf{x}_{i}$. Such a point is called collocation point. In the first works regarding the SPH, a point was included in the stencil of derivative approximations if its distance from the collocation point was less than the smoothing length $h$. This approach has been followed in many other works about SPH-derived methods, such as the RKPM, the CSPm and the MSPH.

In the local max-ent approach, the selection of the stencil was determined by the choice of a parameter $\beta$, that was chosen in order to enforce a condition of locality of the method. In its development, the so called second-order max-ent approach, the shape functions were local automatically, and also the particles included in the stencil were chosen consequently.

Is the Generalized Finite Difference Method, as shown in (Benito et al., 2007), the authors state that an optimal choice of the stencil in a 2d case is the so called "star criterion", that is, the stencil of a collocation point is selected by choosing the two closest particles for each quadrant, independently of the their distance from the collocation point itself, as shown in figure 3.1

When there is not a sufficient number of particles in each quadrant, the missing particles can be supplemented from the other quadrants. This criterion has a general application, but is not precise in the case of structured distribution. Moreover, it can occur the case when all the particles selected by the star criterion are aligned on a cross. In this case it can be shown that the matrix $\mathbf{A}$ of equation (1.60) cannot be inverted.


Figure 3.1: The star criterion

### 3.4 Searching algorithm of neighbour particles

In each meshless method developed in the literature a deep attention has to be given to the algorithm of neighbour particle selection, since it is often the most time consuming part of the whole algorithm, especially when explicit time integration is performed.

In the present section we discuss the algorithm used in the MFPM for the construction of the linear differential operator presented in Sections 3.1 and 3.2 with special attention to the computational cost required in the different phases of the procedure.

In particular, the algorithm takes in input a node distribution and returns the discrete differential operators proposed in the previous section. For both the original and the novel formulation the searching algorithm procedure is exactly the same. The steps composing the particles searching algorithm is then, for each collocation point $\mathbf{x}_{i}$ :

1. all other nodes are reordered on the base of their distance from $\mathbf{x}_{i}$;
2. the 9 particles closest to $\mathbf{x}_{i}$ are chosen;
3. the matrix $\mathbf{A}_{i}$ is built;
4. the matrix $\mathbf{A}_{i}$ is preconditioned on the base of the particle distance from $\mathbf{x}_{i}$, following the idea shown in Ding et al. (2004a,b);
5. the condition number of $\mathbf{A}_{i}$ is computed and compared to a predefined threshold value;
6. if the condition number is higher then the threshold value, the number of neighbour nodes is increased, and phases 3,4 , and 5 are repeated;
7. the matrix $\mathbf{A}_{i}$ is inverted;

Table 3.2: Computational cost for algorithm 1 used on a distribution of 6561 collocation points

| Phase | Computational cost [s] | Percentage [\%] |
| :---: | :---: | :---: |
| 1 | 4.75 | 31 |
| 2 | 0.125 | 0.83 |
| 3 | 0.703 | 4.67 |
| 4 | 0.0892 | 0.59 |
| 5 | 7.65 | 50.85 |
| 6 | 0.409 | 2.71 |
| 7 | 0.344 | 2.28 |
| 8 | 0.975 | 6.48 |

Table 3.3: Computational cost for algorithm 1 used on a distribution of 16461 collocation points

| Phase | Computational cost [s] | Percentage [\%] |
| :---: | :---: | :---: |
| 1 | 21.8 | 42.57 |
| 2 | 0.356 | 0.69 |
| 3 | 2.21 | 4.32 |
| 4 | 0.245 | 0.48 |
| 5 | 20.21 | 39.47 |
| 6 | 0.98 | 1.91 |
| 7 | 0.92 | 1.80 |
| 8 | 4.48 | 8.75 |

8. the $i$-th row the discrete differential operator is built.

The computational cost connected to the different phases depends strongly on the strategies adopted especially during the phases 1 and 2, and the specific in-built MATLAB functions used for phase 5. In particular, this primitive, rough algorithm, for each collocation points reorders all nodes of the domain in terms of distance, and uses the MATLAB in-built function condest. For the algorithm just described the computational costs are reported in Table 3.2

The same algorithm tested on a distribution of 16461 nodes gives the results shown in Table 3.3

It is evident from a first glance to Tables 3.2 and 3.3 that the most costly phases are the phases 1 and 5 . The research algorithm is then significantly improved properly modifying these two phases.

The improvement of phase 1 is made through a different searching algorithm, used in many SPH codes and applications: it consists in dividing the domain in a predefined number of square subodomains (depending on the amount of collocation points, in such a way that each square subdomain contains limited number of nodes). Therefore the research of neighbour particles is made on only the square in which the collocation node is located, and in the 8 adjacent squares. This process obviously reduces the time required for pre-ordering and selecting the neighbour particles.

For what concerns the computational cost of the computation of the condition number of

Table 3.4: Computational cost for algorithm 2 used on a distribution of 6561 collocation points

| Phase | Computational cost [s] | Percentage [\%] |
| :---: | :---: | :---: |
| 1 | 0.435 | 26.12 |
| 2 | 0.0362 | 2.17 |
| 3 | 0.271 | 16.27 |
| 4 | 0.0643 | 3.86 |
| 5 | 0.270 | 16.21 |
| 6 | 0.266 | 15.97 |
| 7 | 0.201 | 12.06 |
| 8 | 0.122 | 7.33 |

Table 3.5: Computational cost for algorithm 2 used on a distribution of 16461 collocation points

| Phase | Computational cost [s] | Percentage [\%] |
| :---: | :---: | :---: |
| 1 | 1.08 | 27.17 |
| 2 | 0.084 | 2.11 |
| 3 | 0.63 | 15.85 |
| 4 | 0.146 | 3.67 |
| 5 | 0.641 | 16.13 |
| 6 | 0.708 | 17.81 |
| 7 | 0.401 | 10.09 |
| 8 | 0.285 | 7.17 |

the matrix $\mathbf{A}_{i}$, it is sufficient to our scope to approximate the condition number, since we only ask to this phase to understand if the matrix $\mathbf{A}_{i}$ is invertible or not, and this information is given also with a non extremely precise condition number. Therefore we replace the MATLAB function condest with the more efficient MATLAB in-built function cond. For technical differences between the two functions we refer to the MATLAB guide.

After the modification of the algorithm, the total computational cost involved is reported in Table 3.4 for 6561 nodes, and in Table for 3.5 for a distribution of 16461 nodes.

From the comparison of Tables 3.2 and 3.4 it is evident the saving of time in the second case.

## Chapter 4

## Applications of the Modified Finite Particle Method to linear elasticity

In the present chapter we apply the Modified Finite Particle Method to linear elasticity. In a first part of the chapter we present the continuous equations that model the behaviour of an elastic body, then we show how such equations are discretized using a Modified Finite Particle Method, in particular we show how do we handle the dynamic term, and finally we show some applications in 2D and 3d statics, 1d and 2d dynamics.

### 4.1 Linear elasticity

In the following we introduce the linear elastic problem in the three-dimensional space and show how it can be formulated with the Modified Finite Particle Method.

We consider an elastic body on a domain $\Omega$, subjected to internal forces $\boldsymbol{b}=\boldsymbol{b}(\boldsymbol{x}, t)$, prescribed displacements $\overline{\boldsymbol{s}}=\overline{\boldsymbol{s}}(\boldsymbol{x}, t)$ on the Dirichlet boundary $\Gamma_{D}$, and prescribed tractions $\overline{\boldsymbol{t}}=\overline{\boldsymbol{t}}(\boldsymbol{x}, t)$ on the Neumann boundary $\Gamma_{N} . \Gamma_{D}$ and $\Gamma_{N}$ are such that

$$
\left\{\begin{array}{l}
\Gamma_{D} \cup \Gamma_{N}=\Gamma \\
\Gamma_{D} \cap \Gamma_{N}=\emptyset
\end{array}\right.
$$

where $\Gamma$ is the whole boundary of $\Omega$.
The equations governing the problem are

$$
\left\{\begin{array}{rlrll}
\rho \frac{\partial^{2} \boldsymbol{s}}{\partial t^{2}} & =\nabla \cdot \sigma \cdot \boldsymbol{\sigma}+ & & x \in \Omega  \tag{4.1}\\
\boldsymbol{\sigma} \boldsymbol{n} & = & \overline{\boldsymbol{t}}(t) & & x \in \Gamma_{N} \\
\boldsymbol{s} & = & \overline{\boldsymbol{s}}(t) & & x \in \Gamma_{D} \\
\left.\boldsymbol{s}\right|_{t=0} & = & s_{0}(x) & & x \in \Omega \\
\left.\frac{\partial \boldsymbol{s}}{\partial t}\right|_{t=0} & & & \dot{s}_{0}(x) & \\
x \in \Omega
\end{array}\right.
$$

where $\rho$ is the mass density of the material; $\boldsymbol{n}$ is the outward normal vector at the boundary, $\boldsymbol{s}=\boldsymbol{s}(\boldsymbol{x}, t)$ is the vectorial displacement field, whose components are $u=u(\boldsymbol{x}, t), v=v(\boldsymbol{x}, t)$, and $w=w(\boldsymbol{x}, t) ; \boldsymbol{\sigma}=\mathbb{C}(\nabla \boldsymbol{s})^{S}$ is the symmetric Cauchy stress tensor. $\mathbb{C}$ is the fourth order linear elastic isotropic tensor, whose components are

$$
\begin{equation*}
\mathbb{C}_{i j k l}=\lambda \delta_{i j} \delta_{k l}+\mu\left(\delta_{i k} \delta_{j l}+\delta_{i l} \delta_{j k}\right) \tag{4.2}
\end{equation*}
$$

where $\lambda$ and $\mu$ are the Lamé constants, which can be expressed in terms of the Young modulus $E$ and the Poisson ratio $\nu$ as follows:

$$
\begin{equation*}
\lambda=\frac{E \nu}{(1+\nu)(1-2 \nu)} ; \quad \mu=\frac{E}{2(1+\nu)} \tag{4.3}
\end{equation*}
$$

$(\bullet)^{S}$ denotes the symmetric part of a tensor (i.e., $\left.\boldsymbol{A}^{S}=\frac{1}{2}\left(\boldsymbol{A}+\boldsymbol{A}^{T}\right)\right)$.
Making explicit (4.1) with respect to the components of the displacement $u, v$ and $w$, we obtain

$$
\left\{\begin{align*}
\rho u, t t & =(\lambda+2 \mu) u,_{x x}+\mu\left(u,_{y y}+u,_{z z}\right)+(\lambda+\mu)\left(v,_{x y}+w,_{x z}\right)+b_{x}  \tag{4.4}\\
\rho v, t t & =(\lambda+2 \mu) v,_{y y}+\mu\left(v,_{x x}+v,_{z z}\right)+(\lambda+\mu)\left(u,_{x y}+w,_{y z}\right)+b_{y} \\
\rho w, t t & =(\lambda+2 \mu) w,_{z z}+\mu\left(w,_{x x}+w,_{y y}\right)+(\lambda+\mu)\left(u,_{x z}+v,_{y z}\right)+b_{z}
\end{align*}\right.
$$

The subscripts preceded by a comma indicate partial derivative.
The semi-discrete form of system (4.4), after the spatial discretization shown in Chapters 3 , is then

$$
\begin{equation*}
\rho \ddot{\hat{\boldsymbol{s}}}=\hat{\boldsymbol{K}} \hat{\boldsymbol{s}}+\boldsymbol{b} \tag{4.5}
\end{equation*}
$$

being $\hat{\boldsymbol{s}}$ the vector of the particle values of $\boldsymbol{s}(\boldsymbol{x}, t)$. In a more explicit form, equation (4.5) reads

$$
\left(\begin{array}{c}
\rho \ddot{\hat{\tilde{}}}  \tag{4.6}\\
\ddot{\hat{\boldsymbol{v}}} \\
\rho \ddot{\hat{\boldsymbol{w}}}
\end{array}\right)=\left(\begin{array}{lll}
\hat{\boldsymbol{K}}_{11} & \hat{\boldsymbol{K}}_{12} & \hat{\boldsymbol{K}}_{13} \\
\hat{\boldsymbol{K}}_{21} & \hat{\boldsymbol{K}}_{22} & \hat{\boldsymbol{K}}_{23} \\
\hat{\boldsymbol{K}}_{31} & \hat{\boldsymbol{K}}_{32} & \hat{\boldsymbol{K}}_{33}
\end{array}\right)\left(\begin{array}{c}
\hat{\boldsymbol{u}} \\
\hat{\boldsymbol{v}} \\
\hat{\boldsymbol{w}}
\end{array}\right)+\left(\begin{array}{c}
\boldsymbol{b}_{x} \\
\boldsymbol{b}_{y} \\
\boldsymbol{b}_{z}
\end{array}\right)
$$

being $\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{w}}$ the vectors containing the particle values of the scalar fields $u, v$, and $w$. $\hat{\boldsymbol{K}}_{i j}$ are the blocks of $\hat{\boldsymbol{K}}$, reading

$$
\begin{aligned}
& \hat{\boldsymbol{K}}_{11}=(\lambda+2 \mu) \boldsymbol{D}_{x x}+\mu\left(\boldsymbol{D}_{y y}+\boldsymbol{D}_{z z}\right) \\
& \hat{\boldsymbol{K}}_{22}=(\lambda+2 \mu) \boldsymbol{D}_{y y}+\mu\left(\boldsymbol{D}_{x x}+\boldsymbol{D}_{z z}\right) \\
& \hat{\boldsymbol{K}}_{33}=(\lambda+2 \mu) \boldsymbol{D}_{z z}+\mu\left(\boldsymbol{D}_{x x}+\boldsymbol{D}_{y y}\right) \\
& \hat{\boldsymbol{K}}_{12}=\hat{\boldsymbol{K}}_{21}=(\lambda+\mu) \boldsymbol{D}_{x y} \\
& \hat{\boldsymbol{K}}_{13}=\hat{\boldsymbol{K}}_{31}=(\lambda+\mu) \boldsymbol{D}_{x z} \\
& \hat{\boldsymbol{K}}_{23}=\hat{\boldsymbol{K}}_{32}=(\lambda+\mu) \boldsymbol{D}_{y z}
\end{aligned}
$$

In the spirit of collocation methods, in equation (4.5), the rows of $\hat{\boldsymbol{K}}$ corresponding to the boundary particles, and the corresponding terms of $\boldsymbol{b}$ and $\rho \ddot{\hat{\boldsymbol{s}}}$ have to be replaced with the
discrete form of the boundary conditions. In this way we obtain the final form of the discrete elasto-dynamic problem

$$
\begin{equation*}
K \hat{s}=f \tag{4.7}
\end{equation*}
$$

where the components of $\boldsymbol{f}$ are $\rho \ddot{\hat{\boldsymbol{s}}}-\boldsymbol{b}$ for the rows associated to internal particles, and $\overline{\boldsymbol{s}}$ or $\overline{\boldsymbol{t}}$ for the boundary particles, in case of Dirichlet or Neumann boundary conditions, respectively.

For elasto-static applications, time derivatives in Equation (4.7) are zero, and the system can be immediately solved; therefore both internal and external particle values are found simultaneously.

In case of elasto-dynamics, we have first to discretize time derivatives, with a numerical scheme, i.e.

$$
\begin{equation*}
\ddot{\ddot{\boldsymbol{s}}^{n}}=\frac{\hat{\boldsymbol{s}}^{n+1}-2 \hat{\boldsymbol{s}}^{n}+\hat{\boldsymbol{s}}^{n-1}}{\Delta t^{2}} \tag{4.8}
\end{equation*}
$$

where $\Delta t$ is the time step and superscripts refer to time increments (e.g., $\hat{\boldsymbol{s}}^{n}=\left.\hat{\boldsymbol{s}}\right|_{t=t_{n}}$ ). The equations of system (4.7), collocated at internal particles, read

$$
\begin{equation*}
\sum_{j} K_{i j} \hat{s}_{j}^{n}=\rho \frac{\hat{s}_{i}^{n+1}-2 \hat{s}_{i}^{n}+\hat{s}_{i}^{n-1}}{\Delta t^{2}}-b_{i}^{n} \tag{4.9}
\end{equation*}
$$

while the equations collocated at boundary particles, where no time derivatives are involved, are in the form

$$
\begin{equation*}
\sum_{j} K_{i j} \hat{s}_{j}^{n+1}=\bar{u}_{i}^{n+1} \tag{4.10}
\end{equation*}
$$

Equations (4.10) cannot be solved by explicit time integration, since the values of $\hat{s}_{j}^{n+1}$ may depend, in case of Neumann boundary conditions, on the values of the internal particles at the same time step $t_{n+1}$. To overcome this difficulty, we perform a static condensation of $\boldsymbol{K}$, and separate the equations collocated on internal particles from those collocated on the boundary. The degrees of freedom are also separated, and so the final form of (4.5) is

$$
\left(\begin{array}{ll}
\boldsymbol{K}_{I I} & \boldsymbol{K}_{I B}  \tag{4.11}\\
\boldsymbol{K}_{B I} & \boldsymbol{K}_{B B}
\end{array}\right)\binom{\hat{\boldsymbol{s}}_{I}}{\hat{\boldsymbol{s}}_{B}}=\binom{\rho \ddot{\hat{\boldsymbol{s}}}_{I}-\boldsymbol{b}_{I}}{\overline{\boldsymbol{u}}}
$$

where $\overline{\boldsymbol{u}}$ is the vector of the prescribed displacements or of the tractions at the boundary, and $\boldsymbol{K}_{I I}, \boldsymbol{K}_{I B}, \boldsymbol{K}_{B I}, \boldsymbol{K}_{B B}$ are the blocks of the matrix $\boldsymbol{K}$ obtained with reference to the internal and boundary particles.

From the second set of equations of (4.11) we compute

$$
\begin{equation*}
\boldsymbol{s}_{B}=\boldsymbol{K}_{B B}^{-1}\left(\overline{\boldsymbol{u}}-\boldsymbol{K}_{B I} \hat{\boldsymbol{s}}_{I}\right) \tag{4.12}
\end{equation*}
$$

and we substitute it into the first set of equations of (4.11), obtaining

$$
\begin{equation*}
\rho \ddot{\hat{\boldsymbol{s}}}_{I}-\left(\boldsymbol{K}_{I I}-\boldsymbol{K}_{I B} \boldsymbol{K}_{B B}^{-1} \boldsymbol{K}_{B I}\right) \hat{\boldsymbol{s}}_{I}=-\boldsymbol{K}_{I B} \boldsymbol{K}_{B B}^{-1} \overline{\boldsymbol{u}}-\boldsymbol{b}_{I} \tag{4.13}
\end{equation*}
$$

where the amount $\boldsymbol{K}_{I I}-\boldsymbol{K}_{I B} \boldsymbol{K}_{B B}^{-1} \boldsymbol{K}_{B I}$ is the condensed stiffness matrix, namely $\tilde{\boldsymbol{K}}$.
Equations (4.13) form an unconstrained ordinary differential equation system which can
be solved by a suitable time integration scheme, like the explicit one of Equation (4.8). Once the values of the unknown functions are computed from Equation (4.13) at time step $t_{n+1}$, Equation (4.12) can be used to retrieve the values of the functions at the boundary particles.

### 4.2 Numerical tests

In the following we propose a number of applications of the investigated models. First we introduce three challenging elasto-static problems: the classical test of an infinitely extended plate with a circular hole under a uniform remote traction, the problem of the Cook's membrane in 2D plain strain, and a 3D block with a spherical bore stretched on a face. Regarding dynamic problems, we investigate the wave propagation in a two-dimensional bar under a quasi-impulsive load, and a quarter of an annulus under a sinusoidal body load. We solve all these problems with the original MFPM, the novel MFPM, and the GFDM, in order to compare the performances of these methods.

### 4.2.1 Plate with a circular hole

The geometry of this problem is depicted in Figure 4.1, along with its symmetry boundary conditions and applied loads. The radius of the internal hole is $a=0.2$.


Figure 4.1: Plate with a circular hole: model problem including symmetry boundary conditions and applied loads.

The equations that govern the problem are the 2D elasto-staic version of (4.4). Plain strain condition are assumed. The boundary conditions are

$$
\left\{\begin{array}{lllll}
\boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{n}=0 & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=0 & \text { on } \Gamma_{1} \text { and } \Gamma_{4}  \tag{4.14}\\
\boldsymbol{s} \cdot \boldsymbol{n} & =0 & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=0 & \text { on } \Gamma_{2} \text { and } \Gamma_{5} \\
\boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{n}=\sigma_{0} & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=0 & \text { on } \Gamma_{3}
\end{array}\right.
$$

where $\boldsymbol{n}$ is the outward normal, $\boldsymbol{t}$ is the unit vector tangent to the boundary, and $\sigma_{0}$ is the uniform remote traction.

We solve the problem considering a reduced domain, such that on $\Gamma_{3}$ and $\Gamma_{4}$ we impose boundary conditions according to the exact solution, that is, in terms of stresses,

$$
\begin{gather*}
\sigma_{x x}=\sigma_{0}\left[1-\frac{a^{2}}{r^{2}}\left(\frac{3}{2} \cos 2 \theta+\cos 4 \theta\right)+\frac{3 a^{4}}{2 r^{4}} \cos 4 \theta\right]  \tag{4.15a}\\
\tau_{x y}=\sigma_{0}\left[-\frac{a^{2}}{r^{2}}\left(\frac{1}{2} \sin 2 \theta+\sin 4 \theta\right)+\frac{3 a^{4}}{2 r^{4}} \sin 4 \theta\right]  \tag{4.15b}\\
\sigma_{y y}=\sigma_{0}\left[-\frac{a^{2}}{r^{2}}\left(\frac{1}{2} \cos 2 \theta-\cos 4 \theta\right)-\frac{3 a^{4}}{2 r^{4}} \cos 4 \theta\right] \tag{4.15c}
\end{gather*}
$$

where $(r, \theta)$ are the polar coordinates, $\theta$ being measured from the positive $x$-axis counterclockwise.

We now introduce the Stress Intensity Factor (SIF) for this problem, that is the ratio between the maximum value of $\sigma_{x x}$ and the value of the remote traction $\sigma_{0}$. In this case, the analytical solution provides $S I F=3$. We then numerically solve the problem, assuming

$$
\begin{equation*}
E=100000 P a, \quad \nu=0.33, \quad \sigma_{0}=100 P a \tag{4.16}
\end{equation*}
$$

and compare the analytical value of the SIF with the obtained numerical results. The distribution of the $\sigma_{x x}$ stress obtained with the original MFPM and 251001 particles is shown in Figure 4.2. The relative error on the SIF is computed as

$$
\begin{equation*}
\text { err }_{r}=\frac{\left|S I F_{\text {an }}-S I F_{\text {num }}\right|}{\left|S I F_{\text {an }}\right|}=\frac{\left|3-S I F_{\text {num }}\right|}{3} \tag{4.17}
\end{equation*}
$$

and convergence plots referred to the three considered methods are reported in Figure 4.3, where $N$ is the total number of particles used for the numerical solution.

We observe that for this problem all three methods show the same second-order convergence, but the error computed with the original MFPM shows a lower constant. With the other two methods quite similar values of the error are achieved, but the computational cost is significantly reduced, since no Voronoi tessellation of the domain is needed.

In Table 4.1 the make a comparison among the computational costs of the original and novel MFPM for this problem. We notice the significant time reduction in the case of novel MFPM.

We observe that for this problem all three methods show the same second-order convergence, but the original formulation has an higher constant. With the other two methods quite similar values of the error are achieved, and the computational cost is significantly reduced, since no Voronoi tessellation of the domain is needed.

### 4.2.2 The Cook's membrane

The Cook's membrane is a classical benchmark introduced by Cook and Al-Abdulla (1969) to show the performance of plane finite elements in dealing with volumetric locking. The


Figure 4.2: $\sigma_{x x}$ in a square with a central hole obtained with the original MFPM and 251001 particles.
geometry is shown in Figure 4.4.
The data of the problem are $L=48 \mathrm{~m}, H_{1}=44 \mathrm{~m}, H_{2}=16 \mathrm{~m}$.
The equations that govern the problem are the 2D elasto-static version of (4.4). Plain strain conditions are assumed. The boundary conditions are

$$
\left\{\begin{array}{llll}
\boldsymbol{s} \cdot \boldsymbol{n} & =0 & \text { and } & \boldsymbol{s} \cdot \boldsymbol{t}=0  \tag{4.18}\\
\boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{n} & =0 & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=0 \\
\boldsymbol{\sigma} \cdot \boldsymbol{n} & =0 & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=\tau_{0}
\end{array} \quad \text { on } \Gamma_{2} \text { and } \Gamma_{4}\right.
$$

where $\tau_{0}=1 / 16 P a$ is a constant shear stress distribution.

| $\sqrt{N}$ | Computational cost of <br> the Voronoi tessellation <br> and cutting algorithm[s] | Total time of the <br> code-original <br> formulation $[\mathrm{s}]$ | Total time of the <br> code - novel <br> formulation $[\mathrm{s}]$ | time saving [\%] |
| :---: | :---: | :---: | :---: | :---: |
| 11 | $5.810^{-1}$ | $2.1610^{0}$ | $1.5310^{0}$ | 29.17 |
| 21 | $2.0410^{0}$ | $4.8510^{0}$ | $2.7810^{0}$ | 42.68 |
| 41 | $7.5710^{0}$ | $1.4710^{1}$ | $7.1910^{0}$ | 51.02 |
| 81 | $2.9210^{1}$ | $5.8510^{1}$ | $2.7310^{1}$ | 53.31 |
| 161 | $1.1610^{2}$ | $2.7610^{2}$ | $1.5910^{2}$ | 42.27 |
| 321 | $4.6210^{2}$ | $1.7210^{3}$ | $1.4010^{3}$ | 18.87 |

Table 4.1: Comparison of the computational costs between the original and novel MFPM for the problem of Figure 4.1.


Figure 4.3: Logaritmic convergence diagram of the error of the SIF in a square plate with a central hole with respect to the square root of the total number of particles included for the discretization.

We solve the problem both in the case of $\nu=0.33$ and in the case of quasi-incompressible material ( $\nu=0.49$ ).

In Figure 4.5 we show the shear stress distribution in the case of $E=1 P a$ and $\nu=$ 0.33 with the original MFPM and 103041 particles. The convergence of the value of the displacement of reference point $C$ (Figure 4.4) is reported in Figure 4.6.

We observe from Figure 4.6 that the solution seems to converge to similar values with all the methods. Again, the novel MFPM and the GFDM perform in a similar way.

For the case of $\nu=0.49$, the shear stress distribution and the convergence diagram are depicted in Figures 4.7 and 4.8, respectively. The number of particles included in the stencil (that is, the group of particles that contribute to the approximation of derivatives) of the original MFPM is 9 , and it looks sufficient for a good performance of the method. The numerical test performed with the GFDM is carried out with 9 particles in the stencil, selected with the star criterion, as described in Benito et al. (2007), but we observe that for this problem the numerical solution does not converge. The same behaviour is obtained with the novel MFPM, and 9 particles in the stencil. For this reason, in Figures 4.8, the convergence diagram of the novel MFPM is computed including 25 particles in the stencil, leading to a convergent approximation.

In Table 4.2 the computational costs of this problem for the original and novel MFPM are reported. Again, we notice the significant time reduction in the case of novel MFPM. From the last columns of Tables 4.1 and 4.2, we also notice that the percentage reduction of time is higher at lower number of particles. This is due to the fact that the computation cost of the Voronoi tessellation algorithm grows less than the one of other routines present in our code, when the number of particles increases.


Figure 4.4: The Cook's problem: geometry, boundary conditions and applied loads.

| $\sqrt{N}$ | Computational cost of <br> the Voronoi tessellation <br> and cutting algorithm[s] | Total time of the <br> code - original <br> formulation [s] | Total time of the <br> code - novel <br> formulation $[\mathrm{s}]$ | time saving [\%] |
| :---: | :---: | :---: | :---: | :---: |
| 11 | $5.710^{-1}$ | $2.4610^{0}$ | $1.5610^{0}$ | 36.33 |
| 21 | $1.6110^{0}$ | $3.5410^{0}$ | $1.8610^{0}$ | 47.03 |
| 41 | $6.1210^{0}$ | $1.2810^{1}$ | $6.4110^{0}$ | 49.92 |
| 81 | $2.3910^{1}$ | $5.2110^{1}$ | $2.7010^{1}$ | 48.21 |
| 161 | $9.4710^{1}$ | $2.5110^{2}$ | $1.9610^{2}$ | 21.93 |
| 321 | $3.8310^{2}$ | $1.9410^{3}$ | $1.4110^{3}$ | 27.22 |

Table 4.2: Comparison of the computational costs between the original and novel MFPM for the problem of Figure 4.4.

### 4.2.3 Multi-material problems

The Modified Finite Particle Method can be applied also to problem implying different materials. There are two strategies that can be used. One strategy consists in considering the variation of the material parameters in the stiffness matrix of the problem.

In fact, given the equilibrium equation for a static problem

$$
\begin{equation*}
\nabla \cdot \boldsymbol{\sigma}+\mathbf{b}=0 \tag{4.19}
\end{equation*}
$$

and the constitutive relation of the material, in the form

$$
\begin{equation*}
\boldsymbol{\sigma}=\lambda(\mathbf{x})(\operatorname{tr} \boldsymbol{\varepsilon}) \mathbf{I}+2 \mu(\mathbf{x}) \boldsymbol{\varepsilon} \tag{4.20}
\end{equation*}
$$

the equilibrium equation becomes, expressing all the variables in terms of the displacement field $\mathbf{u}$

$$
\begin{equation*}
\nabla \lambda(\nabla \cdot \mathbf{u}) \mathbf{I}+2 \nabla \mu \varepsilon+(\lambda+\mu) \nabla(\nabla \cdot \mathbf{u})+\mu \Delta \mathbf{u}+\mathbf{b}=0 \tag{4.21}
\end{equation*}
$$



Figure 4.5: Shear stress distribution in the Cook's membrane obtained with original MFPM and 103041 particles


Figure 4.6: Convergence of the displacement of the point C in the Cook's membrane in the cases of orginal MFPM, novel MFPM and GFDM.


Figure 4.7: Shear stress distribution in a quasi-incompressible Cook's membrane obtained with the original MFPM and 103041.

This approach is particularly useful when material properties vary continously in the domain. It could be used also when a sudden variation of the material occurs, i.e., when the domain is composed of two parts with different material properties. In this case, however, the algorithm will experience the typical problems of finite difference approaches in dealing with non differentiability points, that are oscillation of the solution.

The second possibility is the multi-patch formulation. Each materia subdomain is considered independently in a first moment, and a stiffness matrix for each patch is built, depending on the material parameters. Then, an assembly procedure is performed, imposing the interface boundary condition, that are the continuity of the outward stresses and the continuity of the displacements. This procedure is more computationally expensive from a coding point of view, but avoids oscillation deriving from the need of computing derivative on discontinous fields.

In Figures 4.10 and 4.11 we show the deformation and the displacement of a multi-material domain (see Figure 4.9) under traction. The material parameters are such that the transversal deformation of both the parts of the domain is the same.

### 4.2.4 Three-dimensional elasticity problem

We study the elasticity of a 3D block under a uniform traction. The geometry of this problem is depicted in Figure 4.12. A uniform normal traction $\sigma_{0}=100 P a$ is applied on the face $x=L$. The data of the problem are: $L=5 m, H=3 m, B=2.5 m, R=2 m$.

The equations governing the problem are the static version of (4.1); the boundary condi-

(a) Complete diagram

(b) Zoom of the right boundary of the diagram

Figure 4.8: Convergence of the displacement of the point C in the quasi-incompressible Cook's membrane in the cases of orginal MFPM, novel MFPM, and GFDM.


Figure 4.9: Multi-patch test


Figure 4.10: Multi-patch test: deformed configuration


Figure 4.11: Multi-patch test: displacement in the $x$-direction
tions are

$$
\left\{\begin{array}{llll}
\boldsymbol{s} \cdot \boldsymbol{n}=0 & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=0 & \text { on the faces } x=0, y=0, \text { and } z=0  \tag{4.22}\\
\boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{n}=0 & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=0 & \text { on the faces } y=B, z=H, \\
& & \text { and on the surface of the bore } \\
\boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{n}=\sigma_{0} & \text { and } & \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{t}=0 & \text { at } x=L
\end{array}\right.
$$



Figure 4.12: Geometry of the parallelepiped with a spherical bore.

In Figure 4.13 we show the stress distributions on the symmetry plane $y=0$ obtained using the novel MFPM with 83730 particles and the corresponding Finite Element overkilled solution ( 250476 nodes) obtained with the software Abaqus. We observe a good agreement of the stress distributions $\sigma_{x x}, \sigma_{z z}$, and $\tau_{x z}$ between our numerical results and the overkilled solution provided by Abaqus.

### 4.3 Applications of the MFPM in dynamics

In the following we show the application of the Modified Finite Particle Method to elastodynamic problems. In particular, we explore the case of a 1d barr under quasi impulsive load at the right side, and study the wave propagation. For this problem we also study the properties of the MFPM


Figure 4.13: Stress distributions of the problem of Figure 4.12 obtained by novel MFPM ( 83730 paricles) and overkilled Abaqus solution (250476 nodes) at $y=0$.

### 4.4 1d dynamics: a bar under quasi-impulsive axial load

In this section we perform an application of the Modified Finite Particle Method for the solution of the problem of a 1D clamped rod with an impulse on its right boundary.

The equation which governs the problem is

$$
\left\{\begin{array}{l}
\frac{\partial^{2} u}{\partial t^{2}}=a^{2} \frac{\partial^{2} u}{\partial x^{2}}  \tag{4.23}\\
u(x=0, t)=0 \\
E A \frac{\partial u}{\partial x}(x=1, t)=F(t)
\end{array}\right.
$$

where $E$ is the Young modulus of the material, $A$ is the normal area of the cross section of the bar, $a$ is the velocity of the sound, and its value is $\sqrt{\frac{E}{\rho}}$, where $\rho$ is the mass density of the material.
$F(t)$ is the forcing function of this problem, that in this case is quasi-impulsive, with expression

$$
F(t)=F_{0} e^{-b\left(t-t_{0}\right)^{2}}
$$

. The temporal amplitude of this expression depends on the magnitude of the term $b$.
The spatial discretization of the present problem is performed in accordance with the 1D version of the MFPM. In particular, before solving the problem, we discuss the Fourier analysis of this kind of equations, in order to see how the MFPM approximates the eigenfrequencies and the eigenfunctions of the continuum problem, which, from a numerical point of view, depend on the eigenvalues and eigenvectors of the numerical stiffness matrix; then we study the dispersion relation and the stability of the numerical system of equations, depending on the choice of the spatial and temporal discretization. Finally, we solve the equation and draw a convergence diagram of the error.

### 4.4.1 Fourier analysis of the wave propagation problem

A useful instrument for the numerical analysis of dynamic problems is the Fourier analysis, that is the decomposition of the solution in its harmonic components. To find the different components, we write a problem which is said the eigenvalues and eigenfunctions problem, the last being the non trivial function that solve the equation of wave propagation

The equation that models the dynamics of a 1d extensional bar is

$$
\begin{equation*}
\ddot{u}=u^{\prime \prime} \tag{4.24}
\end{equation*}
$$

where we consider that the propagation velocity is unitary. We reduce to the case of harmonic solutions, and therefore we write the solution in terms of sinusoidal components.

$$
\begin{equation*}
u(x, t)=\bar{u}(x) g(t)=\bar{u} \sin (\omega t) \tag{4.25}
\end{equation*}
$$

Remark. Note that when we do the hypothesis of a temporal function $g(t)=\sin (\omega t)$, we are reducing to the particular case in which the function $g(t)=\sin (\omega t)$ respects the initial conditions. Anyway, such procedure is general, so that we can include all possible initial con-
ditions if only we change the form of the function $g(t)$. Moreover, we can consider also the case in which the solution is not harmonic. In fact, because of the linearity of the problem, any function $g(t)$ can be seen as the sum of harmonics.

Introducing (4.25) in the Equation (4.24), we obtain

$$
\begin{equation*}
\left(\bar{u}^{\prime \prime}+\omega^{2} \bar{u}\right) \sin (\omega t)=0 \tag{4.26}
\end{equation*}
$$

that has not only the trivial solution $\bar{u}=0$, but also infinite solutions that respect th equation

$$
\begin{equation*}
\bar{u}^{\prime \prime}+\omega^{2} \bar{u}=0 \tag{4.27}
\end{equation*}
$$

Equation (4.27) is known as the Helmotz equation, and is the "spectral equation" for an extensional bar. Its general solution is

$$
\begin{equation*}
\bar{u}_{n}=A \sin \left(\omega_{n} t+\phi\right) \tag{4.28}
\end{equation*}
$$

where the parameters $\omega_{n}$ depend on the boundary conditions.
Let's consider the same problem from a discrete point of view. After the MFPM discretization procedure, the equation (4.24) becomes

$$
\begin{equation*}
\ddot{\mathrm{u}}=\mathbf{K u} \tag{4.29}
\end{equation*}
$$

As earlier, according with a particular set of initial conditions, we write

$$
\begin{equation*}
\mathbf{u}=\hat{\mathbf{u}} \sin (\omega t) \tag{4.30}
\end{equation*}
$$

so that we obtain

$$
\begin{equation*}
\left(\omega^{2} \mathbf{I}+\mathbf{K}\right) \hat{\mathbf{u}} \sin (\omega t)=0 \tag{4.31}
\end{equation*}
$$

This equation has the structure of a classical eigenvalue problem, where the amounts $-\omega_{n}^{2}$ are the eigenvalues of the matrix $\mathbf{K}$.

### 4.4.2 Solution of the eigenvalue problem for the clamped bar

Here we specialize what we have just explained to the case of a clamped bar. We consider the problem (4.23) and perform the eigenvalue problem, first in the continuum, then in the discrete form.

From the imposition of the homogeneous boundary conditions (bar clamped on a edge, stress-free on the other edge), the particular expression for the solution of the Helmotz equation (see (4.27)) is

$$
\begin{equation*}
\bar{u}=A \sin \left(\omega_{n} t\right) \tag{4.32}
\end{equation*}
$$

where


Figure 4.14: Ratio between the eigenvalues of the discrete problem and the ones of the continuum problem

$$
\begin{equation*}
\omega_{n}=\frac{\pi}{2}(2 n-1) \tag{4.33}
\end{equation*}
$$

The values of $-\omega_{n}^{2}$ are the analytical eigenvalues of the elastic problem, and the functions $\bar{u}_{n}$ are the eigenvectors (or the eigenmodes).

Now we perform the discrete procedure mentioned before. A good test for the numerical method is to understand how it reproduces the frequencies and the eigenmodes.

The search for eigenvalues in the discrete problem is performed considering an approximation of the derivatives involving both three and five particles. In the second case, we consider all the combination of constant, linear, quadratic and cubic projection functions.

### 4.4.3 Approximation of the dispersion relation

The solution of every linear partial differential equation can be decomposed in the sum of exponential function with complex exponential of the type

$$
\begin{equation*}
e^{i(\omega t-\boldsymbol{\xi} \cdot \mathbf{x})} \tag{4.34}
\end{equation*}
$$

where $\boldsymbol{\xi}$ is the wave vector, which components are the spatial frequency of the solution in the direction of the axes.

In 1D case, the vector $\boldsymbol{\xi}$ reduces to a scalar, namely $\xi$.
If we introduce this solution in the partial differential equation, we will obtain a relation between the temporal frequency $\omega$ and the wavenumber $\xi$. If we assume real values for the wavenumbers, from the dispersion relation we obtain a value for the temporal frequency. In particular:

- if this value is real, we have that the solution is conservative, that is that the eigenmode with spatial frequency $\xi$ oscillates in time;
- if $\omega=\omega(\xi)$ has a positive imaginary part, the solution dampens;
- if $\omega=\omega(\xi)$ has a negative imaginary part, the solution increases its amplitude in time

Another important aspect that we can understand from the dispersion relation is the velocity of propagation of the waves. In a non dispersive problem, all the waves propagate at the same velocity, and we say that this is a non dispersive problem; otherwise we have that each eigenmode propagates at a different velocity, so that the profile of the solution changes in time. The velocity of propagation is called group velocity, and we calculate it

$$
\begin{equation*}
c=\frac{\partial \omega}{\partial \xi} \tag{4.35}
\end{equation*}
$$

In the case of extensional bar, by introducing Equation (4.34) in (4.24), we obtain

$$
\begin{equation*}
\omega^{2}=\xi^{2} \tag{4.36}
\end{equation*}
$$

The group velocity, using (4.35), is then $c=1$. The dynamics of an extensional bar is a non dispersive problem.

The same thing does not happen in the discrete form of the problem; in fact, after the MFPM discretization and the choice of a finite difference scheme for the temporal advance, we set, for a generic particle

$$
\begin{equation*}
u_{h}^{n}=u_{0} e^{i\left(\omega t_{n}-\xi x_{h}\right)} \tag{4.37}
\end{equation*}
$$

and consider that $t_{n}=t+\Delta t_{n}$ and $x_{i}=x+\Delta x_{i}$, after some manipulation we can finally write

$$
\begin{equation*}
\omega=\frac{1}{\Delta t} \cos ^{-1}\left(1+\frac{\Delta t^{2}}{2} \sum_{j=h-n p}^{h+n p} m_{j} e^{-i \xi \Delta x_{j}}\right) \tag{4.38}
\end{equation*}
$$

That, for small values of $\omega \Delta t$ and $\xi \Delta x_{i}$, collapses into the continuum wave dispersion relation.

The coefficients $m_{j}$ are the superdiagonal terms, the diagonal term and the subdiagonal terms of a generic row of the matrix $\tilde{\mathbf{K}}$. Real roots, or real part of complex roots of the wave dispersion relation imply wave propagation; imaginary roots, or imaginary part of complex roots, imply amplification or reduction of the wave.

We see from Equation (4.38) that in general a discrete system is dispersive and also diffusive, that means that the discrete solution have both a reduction of amplitude, connected to the imaginary part of the exponential; and that waves do not propagate at the same velocity, and it depends on the fact that the group velocity is a function of the wavenumber. For this reason we sometimes observe some harmonics that should not be seen in the solution of impulsive problems; it depends on the fact that in the solution some harmonics propagate at a different velocity.

The particular case is when we have a uniform distribution of particles in the domain. In this case the imaginary part of the roots of the equation (4.38) is null, and we have only the phenomenon of dispersion.

In figure 4.15 we can see the discrete dispersion relation in the case of approximation of the derivatives with three particles and five particles (with the different projection functions), for one hundred and one particles and a time step of $10^{-5}$


Figure 4.15: Wave dispersion relation in the continuum and with the spatial a temporal discretization

## Application

In this section, we solve the problem (4.23) for the cases in which $F(t)=-100 e^{-b(t-0.5)^{2}}$. We solve this partial differential equation with $b=100$ and $b=1000$, because the integral ${ }^{1}$ of the two functions have a different harmonic content.

The numerical solution of this problem has been compared with an analytical solution at $T=3$. At that time, the normal force is

$$
\begin{equation*}
N(t=3 s)=100 e^{-b(0.5-x)^{2}} \tag{4.39}
\end{equation*}
$$

The error has been computed as

$$
\begin{equation*}
e r r_{2}=\frac{\left\|\mathbf{u}_{e x}-\mathbf{u}\right\|_{2}}{\left\|\mathbf{u}_{e x}\right\|_{2}} \tag{4.40}
\end{equation*}
$$

The calculation has been performed for a temporal step $\Delta t=10^{-5}$, for a different number of particles. The diagram of convergence of the error is shown in Figure 4.16

As we can see, the case $b=100$ is better reproduced than the case of $b=1000$, where we can see the wave dispersion. As a confirm of that, we show in Figures 4.17 the numerical solution of the problem for 101 particles in the two cases, for the same time step.

### 4.5 2D dynamics

In the present section we the Modified Finite Particle Method to the dynamics of 2D bodies: in particular we first study a 2D bar clamped on the left edge under an impulsive traction on its right side, and study the stress wave propagation; then we study the dynamics of a quarter of annulus under sinusoidal body load. FOr both cases we compare the original and

[^0]

Figure 4.16: Error diagram of the problem (4.23)
novel MFPM formulation, and make also a comparison with the Generalized Finite Difference Method.

### 4.5.1 Dynamics of a 2D bar under quasi-impulsive load

We now consider a two-dimensional bar under a quasi-impulsive load. The geometry and the boundary conditions are depicted in Figure 4.18, where $L=1 \mathrm{~m}$ and $H=0.2 \mathrm{~m}$.

The equations that govern the problem are the 2D plane strain version of (4.4); the boundary conditions are

$$
\left\{\begin{array}{rlrlrl}
u & =0 & & \text { and } & & v
\end{array}=0 \begin{array}{lrl} 
& & \text { on } \Gamma_{1}  \tag{4.41}\\
\sigma_{y y} & =0 & \\
\sigma_{x x} & =\sigma(t) & \\
\text { and } & & \text { and }
\end{array}\right.
$$

where $\sigma(t)=\sigma_{0} \exp \left(-b\left(t-t_{0}\right)^{2}\right)$ is the quasi-impulsive load on the right end of the bar; the test has been performed considering a Poisson ratio equal to zero, so to reproduce a onedimensional test. We also set $E=100 P a$ and $\rho=100 \mathrm{Kg} / \mathrm{m}^{3}$. For this test an analytical solution is available for $\sigma_{x x}(x, y)$, since the analytical propagation velocity $c=\sqrt{E / \rho}$ is known.

The other data for this problem are:

$$
\begin{equation*}
\sigma_{0}=-100 P a, \quad b=1001 / s, \quad t_{0}=0.3 \mathrm{~s} \tag{4.42}
\end{equation*}
$$

The numerical results of $\sigma_{x x}$ obtained using a time step $\Delta t=10^{-4} s$ are compared with the analytical solution after 2.5 s from the impulse, so that the analytical reference solution is

$$
\begin{equation*}
\left.\sigma_{x x}(x, y)\right|_{t=2.8}=-\sigma_{0} \exp \left(-b(x-0.5)^{2}\right) \tag{4.43}
\end{equation*}
$$



Figure 4.17: Numerical solutions

The relative error is computed as

$$
\begin{equation*}
e r r_{r}=\frac{\left\|\boldsymbol{\sigma}_{x x, a n}-\boldsymbol{\sigma}_{x x, n u m}\right\|}{\left\|\boldsymbol{\sigma}_{x x, a n}\right\|} \tag{4.44}
\end{equation*}
$$

We show in Figure 4.20 the convergence of the error for this test. We observe that both the novel MFPM and the GFDM behave in the same way until the computation carried out with 201x201 particles, where the GFDM exhibits numerical instability, while the novel MFPM does not. The original MFPM remains stable until 201x201 particles, but with a higher constant of the error.


Figure 4.18: Geometry, boundary conditions and loads of the bar under quasi-impulsive load,

(a) $t=0.56 \mathrm{~s}$

(b) $t=1.11 \mathrm{~s}$

(c) $t=2.76 \mathrm{~s}$

Figure 4.19: The stress component $\sigma_{x x}$ in the bar during some time instants, obtained with the novel MFPM and 2121 particles.


Figure 4.20: Convergence diagram for the bar under quasi-impulsive load.

### 4.5.2 Quarter of annulus under a sinusoidal body load

In this section we solve the elasto-dynamic problem studied in Auricchio et al. (2012). The geometry of this problem is depicted in Figure 4.21. The structure is clamped on all its boundary, and undergoes a sinusoidal body load. For the internal particles Equations (4.4) hold. The internal radius is $r=1 \mathrm{~m}$, while the external one is $R=4 \mathrm{~m}$.

The internal body loads and the initial conditions have been manufactured so that the analytical solution for the displacements $u$ and $v$ is

$$
\begin{equation*}
u(x, y, t)=v(x, y, t)=\frac{1}{100} x y\left(x^{2}+y^{2}-16\right)\left(x^{2}+y^{2}-1\right) \sin (2 \pi t) \tag{4.45}
\end{equation*}
$$

The relative error

$$
\begin{equation*}
e r r_{r}=\frac{\left\|u_{a n}-u_{n u m}\right\|}{\left\|u_{a n}\right\|} \tag{4.46}
\end{equation*}
$$

has been computed at time $t=1.75 \mathrm{~s}$. The time step used for the analysis is $\Delta t=10^{-4} \mathrm{~s}$. In Figure 4.22 we show the rate of convergence of the error and we observe that the second-order accuracy of the method is confirmed. We remark that in this example the GFDM and the novel MFPM perfectly coincide.


Figure 4.21: Domain of the quarter of annulus with sinusoidal body load.


Figure 4.22: Convergence diagram of the error for the quarter of annulus under sinusoidal body load.

## Chapter 5

## Modified Finite Particle Method applied to quasi-incompressible materials


#### Abstract

In this chapter we apply the Modified Finite Particle Method on incompressible and quasiincompressible elasticity problems. In particular, the displacement-based formulation is investigated in the limit of incompressibility ( $\nu \rightarrow 0.5$ ), and then the Stokes equations for full incompressible solids are investigated. In the field of Finite Difference Method it is well known (Strikwerda, 1984) that the classical discretization of the Stokes Equation on non-staggered grids leads to spurious numerical errors, known as checkerboard instability of pressure. These oscillations are due to the non satisfaction of the inf-sup condition, first studied by Brezzi (1974) in the field of Finite Element Method. For this reason, in order to discretize the Stokes problem on non-staggered grids (and then on meshless methods, where staggered grids are not permitted), some different formulations have to be introduced. In particular, the incompressibility constraint equation is replaced by a derived equation, called Pressure Poisson Equation, in which the respect of the inf-sup condition is not requested. However, on this formulation it is not evident which set of boundary condition is needed. A significant contribution to this discussion has been given in (Gresho and Sani, 1987; Sani et al., 2006), where the problem is faced using a weak formulation.

The chapter is organized as follows: in Section 1 we recall the equations that describe the statics of solids, first in the compressible form, and then in the limit of incompressibility and, finally, we introduce the Stokes Equations for full incompressibility. In Section 2 we introduce the Poisson Pressure Equation formulation, and the problem of the correct choice of boundary conditions, and in Section 3 we apply the Modified Finite Particle Method on a benchmark problem, using the formulations discussed in Section 2.


### 5.1 Governing equations

In the following we introduce the equations that discribe the equilibrium an elastic, incompressible body. We first introduce the equations in the classical displacement-based formulation,
then we switch to a mixed, displacement-pressure based formulation, in order to fully enforce the incompressibility constraint. In the applications, we will show that the limit to incompressibility of the displacement-based formulation leads to numerical problems, that is the main cause for which the displacement-pressure formulation is introduced.

We consider an elastic body within a domain $\Omega$, subjected to internal forces $\mathbf{b}=\mathbf{b}(\mathbf{x}, t)$, prescribed displacements $\overline{\mathbf{u}}=\overline{\mathbf{u}}(\mathbf{x}, t)$ on the Dirichlet boundary $\Gamma_{D}$, and prescribed tractions $\overline{\mathbf{t}}=\overline{\mathbf{t}}(\mathbf{x}, t)$ on the Neumann boundary $\Gamma_{N}$. Boundaries $\Gamma_{D}$ and $\Gamma_{N}$ are such that

$$
\left\{\begin{array}{l}
\Gamma_{D} \cup \Gamma_{N}=\Gamma \\
\Gamma_{D} \cap \Gamma_{N}=\emptyset
\end{array}\right.
$$

where $\Gamma$ is the whole boundary of $\Omega$.
The equations governing the problem are

$$
\begin{cases}\nabla \cdot \boldsymbol{\sigma}+\mathbf{b}=\mathbf{0} & \text { for } \mathbf{x} \in \Omega  \tag{5.1}\\ \boldsymbol{\sigma} \mathbf{n}=\overline{\mathbf{t}}(t) & \text { for } \mathbf{x} \in \Gamma_{N} \\ \mathbf{u}=\overline{\mathbf{u}}(t) & \text { for } \mathbf{x} \in \Gamma_{D}\end{cases}
$$

where $\rho$ is the mass density of the material, $\mathbf{a}$ is the material acceleration, $\mathbf{n}$ is the outward normal vector at the boundary, $\mathbf{u}=\mathbf{u}(\mathbf{x}, t)$ is the vectorial displacement field; $\sigma=\mathbb{C}(\nabla \mathbf{u})^{S}$ is the symmetric Cauchy stress tensor. We use the notation $(\bullet)^{S}$ to denote the symmetric part of a tensor (i.e., $\left.\mathbf{A}^{S}=\frac{1}{2}\left(\mathbf{A}+\mathbf{A}^{T}\right)\right)$. The fourth order linear elastic isotropic tensor $\mathbb{C}$ is expressed in index notation as follows

$$
\begin{equation*}
\mathbb{C}_{i j k l}=\lambda \delta_{i j} \delta_{k l}+\mu\left(\delta_{i k} \delta_{j l}+\delta_{i l} \delta_{j k}\right) \tag{5.2}
\end{equation*}
$$

where $\lambda$ and $\mu$ are the Lamé constants, which can be expressed in terms of the Young modulus $E$ and the Poisson ratio $\nu$ :

$$
\begin{equation*}
\lambda=\frac{E \nu}{(1+\nu)(1-2 \nu)} ; \quad \mu=\frac{E}{2(1+\nu)} \tag{5.3}
\end{equation*}
$$

The condition of incompressibility is imposed when the Poisson ratio $\nu$ is set to 0.5 . Unfortunately, when $\nu$ approaches 0.5 , the parameter $\lambda$ tends to infinity, leading to an ill conditioned discrete system of equations, with consequent degradation of the solution (Chi et al., 2014). Therefore, a different formulation is needed where the incompressibility constraint is enforced in a different way.

For an incompressible body, the constitutive relation is modified in the form

$$
\begin{equation*}
\boldsymbol{\sigma}=-p \mathbf{I}+2 \mu \varepsilon \tag{5.4}
\end{equation*}
$$

where $p$ is the pressure, considered, as usual in the fluid-dynamic literature, positive in case of compression. $\mathbf{I}$ is the identity tensor, $\mu$ is the second Lamé constant and $\varepsilon$ is the symmetric part of the gradient of the vector $\mathbf{u}=\mathbf{u}(\mathbf{x})$.

By replacing (5.4) into the first equation of system (5.1), we obtain

$$
\begin{equation*}
-\nabla p+\mu \Delta \mathbf{u}=-\mathbf{b} \tag{5.5}
\end{equation*}
$$

where the incompressibility constraint

$$
\begin{equation*}
\nabla \cdot \mathbf{u}=0 \tag{5.6}
\end{equation*}
$$

is introduced.
Eqns. (5.5) and (5.6) are known as the Stokes equations in primitive variables $(\mathbf{u}, p)$, and describe the dynamics of fully incompressible bodies. They have to be completed with suitable boundary conditions, that can be Dirichlet boundary conditions (when the boundary displacement is known), or Neumann boundary conditions (when the boundary traction is known).

### 5.2 Classical approaches for incompressibility

The discretization of Eqns. (5.5) and (5.6), performed using the same spatial discretization for $\mathbf{u}$ and $p$, leads to a well known instability of the pressure field, due to the non satisfaction of the so-called inf-sup condition (Brezzi and Fortin, 1991). This means that alternative formulations have to be introduced in order to overcome this numerical difficulty.

In the Finite Element Method, the classical way to overcome pressure instability is the use of different interpolations for the velocity and pressure fields, the first being discretized using quadratic elements (i.e. six-nodes triangles), while the pressure is discretized using linear interpolation. In this way, the respect of the LBB condition is ensured, and spurious oscillations of the pressure are avoided.

In the field of collocation methods, in particular in the Finite Difference Method, the standard method to satisfy the LBB condition is the use of staggered grids, called also MAC grids (Harlow et al., 1965) (see Figure 6.1). This kind of grids, however, require rectangular domains and regular node distribuutions, and therefore they are not suitable for meshless methods, where, in general, non regular distributions of points are permitted.

In order to solve the Stokes problem on non-staggered grids, many different formulations have been introduced in the literature (Gresho and Sani, 1987; Sani et al., 2006; Wang and Liu, 2000; E and Liu, 2003). In particular, the previous works are concentrated on whether boundary conditions are required or not, at a discrete level, for the incompressibility equations. In fact the constraint equation holds both on the interior and on the boundary of the domain, and then, no additional boundary condition is required.

A reference work regarding this discussion is the one by Sani et al. (2006), in which a deep mathematical analysis is done, in the context of the weak formulation. In particular, the analysis is done on the so called Stokes problem with the Poisson Pressure Equation, where the constraint equation of incompressibility is replaced by an equation on the pressure obtained applying the divergence operator on the equations of equilibrium (5.5).

$$
\begin{equation*}
\nabla \cdot(-\nabla p+\mu \Delta \mathbf{u})=-\nabla \cdot \mathbf{b} \tag{5.7}
\end{equation*}
$$



Figure 5.1: A staggered grid
that is, separating the different components at the left-hand side

$$
\begin{equation*}
-\Delta p+\mu \nabla \cdot(\Delta \mathbf{u})=-\nabla \cdot \mathbf{b} \tag{5.8}
\end{equation*}
$$

Eqn. (6.9) is referred, in (Sani et al., 2006), as the Consistent Poisson Pressure Equation (CPPE). Changing the order between the Laplacian and divergence operators (that are commutative differential operators) in the term $\mu \nabla \cdot(\Delta \mathbf{u})$ we obtain $\mu \Delta(\nabla \cdot \mathbf{u})$ that is evidently zero due to the incompressibility equation. This permits to simplify Eqn. (6.9), obtaining the so called Simplified Pressure Poisson Equation.

$$
\begin{equation*}
\Delta p=\nabla \cdot \mathbf{b} \tag{5.9}
\end{equation*}
$$

In (Sani et al., 2006) the discussion is performed in particular on which boundary conditions are required for the solution of the incompressibility problem using the Consistent Pressure Poisson Equation and the Simplified Pressure Poisson Equation. In particular, using the CPPE, no boundary conditions are required for the constrain equation; on the constrary, using the SPPE, a Neumann boundary condition for the pressure is required, obtained projecting the equilibrium equation on the outward normal at the boundary, that is

$$
\begin{equation*}
\frac{\partial p}{\partial \mathbf{n}}=(\mu \Delta \mathbf{u}+\mathbf{b}) \cdot \mathbf{n} \tag{5.10}
\end{equation*}
$$

In this paper we solve the Stokes problem using both the Consistent and the Simplified Pressure Poisson Equation, and using the Modified Finite Particle Method to discretize spatial derivatives.

We also solve the incompressibility problem using, instead of boundary condition (5.10), the discretization of the divergence constrain at the boundary. We refer to this possibility as
the SPPE-div formulation.

### 5.3 Applications

In this section we apply the Modified Finite Particle Method to discretize the spatial derivatives of the formulations presented in the previous section: the Consistent Pressure Poisson Equation, the Simplified Pressure Poisson Equation (with the boundary condition for the constraint equation proposed by Sani et al. (2006)), and the SPPE-div. In particular, we test the effectiveness of these formulation on an incompressible square under a vertical body load, clamped on two edges. We test the MFPM on a displacement-based formulation in the limit of incompressibility $(\nu \rightarrow 0.5)$ and then on the mentioned incompressible formulations, and remark that on the incompressible formulations, the incompressibility constraint is not enforced strongly, but through a derived equation. For this reason we investigate, on both problems, how the incompressibility is respected.

### 5.3.1 Square clamped on two edges under a vertical body load

The problem under investigation is a square in the domain $[0,1] \times[0,1]$, under a vertical body load $\mathbf{b}=-80 \mathbf{e}_{2}$, with boundary conditions (see Figure 6.13)

$$
\left\{\begin{array}{rllll}
\mathbf{u}=0 & x=0 & \text { or } & y=0  \tag{5.11}\\
\boldsymbol{\sigma} \mathbf{n}=0 & x=1 & \text { or } & y=1
\end{array}\right.
$$

This problem has been solved in Auricchio et al. (2007) using the stream-function formulation and an isogeometric approach for the spatial discretization. The second Lamé constant is $\mu=40$.


Figure 5.2: Square clamped on two edges under a vertical body load: geometry and boundary conditions
Here we solve this problem using a displacement-based formulation in the limit of incompressibility $(\nu=0.49, \nu=0.499, \nu=0.4999, \nu=0.49999)$ and using the CPPE, SPPE and

SPPE-div.


Figure 5.3: Square clamped on two edges under a vertical body load: values of the divergence of the displacements in the domain

Using CPPE formulation, similarly to the previous case, the incompressibility is not verified, as it can be seen from Figure 5.3, where the values assum by the divergence of displacements is shown, using 10201 particles. For what concerns other formulations, the corresponding convergence diagram of the error is shown in Figure 5.4 for the vertical displacement of the point $B$. We notice that the displacement based formulations have results compatible with the numerical problems of the locking; SPPE formulation also shows no convergence, while the SPPE-div formulation shows convergence even faster than the expected second-order. We remark also that in this case is not possible to compute a 2 nd norm of the error, since we do not have an analytical solution. We only can compute the relative error in some sampling points, as reported in Auricchio et al. (2007).

In Figure 5.5 we show the deformed configuration obtained with MFPM and a displacementbased formulation (58081 nodes) and $\nu=0.4999$. A comparison with Figure 5.6, in which an overkilled deformed structure is shown, highlights that the displacement-based methods, in the limit of incompressibility, suffer from volumetric locking.

### 5.4 Conclusions

In the present chapter we applied the Modified Finite Particle Method to the problem of incompressible elasticity. In particular, some different formulations have been investigated: a displacement-based formulation, in the limit of incompressibility, with $\nu \rightarrow 0.5$, and three different formulations of the Stokes problem. For these formulations, in particular, the incompressibility constrain $(\nabla \cdot \mathbf{u})$ is not imposed strongly, but it is replaced by a derived one, in which the Lapacian operator is applied to pressure. This choice is done to overcome the difficulties related to the non-respect of the inf-sup condition, which results in unphysical oscillations of the pressure field.

Unfortunately, these derived formulation may need some boundary conditions for the constrain equation, that are not needed by the original Stokes problem in the divergence form.


Figure 5.4: Square clamped on two edges under a vertical body load: 2nd norm error using MFPM on a displacement-based formulations, SPPE, and SPPE-div


Figure 5.5: Square clamped on two edges under a vertical body load: deformed configuration obtained with displacement-based formulation and 58081 nodes. The structure exibiths volumetric locking


Figure 5.6: Square clamped on two edges under a vertical body load: deformed configuration obtained with the Gauge method and 58081 nodes.

In this paper we investigate three different fully-incompressible formulations: the so Consistent Pressure Poisson Equation, the Simplified Pressure Poisson Equation and the Simplified Pressure Poisson Equation with the divergence constrain at the boundary. These formulations differ among each other for the boundary conditions imposed on the constrain equation: in the first ccase, according to Sani et al. (2006), no boundary conditions are required; in the second case, where the normal component of the displacement is know, a boundary condition for the pressure is obtained projecting the equilibrium equation on the outward normal; in the SPPE-div, instead, the divergence-free constrain is applied as boundary condition for the Pressure Poisson equation on the whole boundary.

Here we see that for the CPPE formulation, the incompressibility constraint is not respected for both problems under investigation; the SPPE exhibits lower convergence of the error with respect to the expected second order; and finally the SPPE-div formulation exhibits correct second-order accuracy, even if with a high constant of the error. The displacement-based formulations, even if correctly discretized with MFPM, exhibit the numerical pathology of locking.

## Pubblication

An extended version of the present chapter has been published in Asprone et al. (2015).

## Chapter 6

## Full incompressible solids and fluids

### 6.1 Stokes problem: classical formulation

In the present section we introduce the Stokes equations for the solution of problems involving incompressible solids and fluids. In the first part we focus on the mathematical expression of the Stokes equations and give a different interpretation of variables depending on whether the body under consideration is a solid or a fluid. In the second part we describe the commonly used methodologies for the numerical solution of the Stokes problem in the context of the Finite Element Method and of the Finite Difference Method.

The equations that describe the dynamics of an incompressible body are

$$
\left\{\begin{array}{l}
\rho \mathbf{a}=-\nabla p+\mu \Delta \mathbf{u}+\mathbf{b}  \tag{6.1}\\
\nabla \cdot \mathbf{u}=0
\end{array}\right.
$$

where the first equation is the linear equilibrium equation, and the second is the incompressibility constraint. Equations (6.1) have to be completed with suitable boundary and initial conditions.

In Equations (6.1) the variable $\rho$ is the material density, $\mu$ is the shear modulus (that in fluid dynamics assumes the denomination of dynamic viscosity), $p$ is the pressure, assumed positive in compression, and $\mathbf{b}$ is the vector of the internal body loads. The variable $\mathbf{u}$ assumes different physical meanings depending on whether the body under consideration is a solid or a fluid.

In the case of an incompressible solid, $\mathbf{u}$ is the displacement field, and therefore the inertial term $\rho \mathbf{a}$ is expressed as

$$
\begin{equation*}
\rho \mathbf{a}=\rho \frac{\partial^{2} \mathbf{u}}{\partial t^{2}} \tag{6.2}
\end{equation*}
$$

For an incompressible fluid, conversely, the variable $\mathbf{u}$ represents the velocity field, and therefore the inertial term is written as

$$
\begin{equation*}
\rho \mathbf{a}=\rho \frac{\partial \mathbf{u}}{\partial t}+\rho \mathbf{c} \cdot \nabla \mathbf{u} \tag{6.3}
\end{equation*}
$$

The term $\mathbf{c}$ is the relative velocity between the fluid and the reference frame. When we
assume a total Eulerian formulation, $\mathbf{c}=\mathbf{u}$, and therefore the equilibrium equation is modified in the form

$$
\begin{equation*}
\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u} \cdot \nabla \mathbf{u}\right)=-\nabla p+\mu \Delta \mathbf{u}+\mathbf{b} \tag{6.4}
\end{equation*}
$$

that is evidently a non-linear equation.
In the present paper, however, we neglect the non-linear term, since the computational difficulties involved in the solution of the incompressibility problem are still evident also in the linear case.

In the theoretical discussion of next sections, we refer to the interpretation of variables as in the fluid case. Therefore, the set of equations under our attention is:

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{u}}{\partial t}=-\nabla p+\mu \Delta \mathbf{u}+\mathbf{b}  \tag{6.5}\\
\nabla \cdot \mathbf{u}=0
\end{array}\right.
$$

Equations (6.5) are known as Stokes equations in the primitive variables $\mathbf{u}$ and $p$, and correspond to the assumption of highly viscous flows.

### 6.1.1 Classical numerical schemes for the solution of the Stokes problem

The discretization of Equations (6.5), performed using the same spatial discretization for u and $p$, leads to a well known instability of the pressure, known in the literature as checkerboard instability. Such pressure unphysical oscillation can be avoided when a numerical condition, known as LBB condition, or inf-sup condition, is respected.

In the Finite Element Method, the classical way to overcome pressure instability is the use of different interpolations for velocity and pressure fields: as an example, the velocity can be discretized using quadratic interpolation, while the pressure can be discretized using linear interpolation. This choice of interpolations ensures the respect of the inf-sup condition and therefore the spurious pressure oscillations are avoided.


Figure 6.1: A staggered grid

The problem of checkerboard instability arises also in the context of collocation methods, as shown in Strikwerda (1984). In particular, in the Finite Difference Method this numerical difficulty is solved using staggered grids, called also MAC grids (Harlow et al., 1965), where horizontal and vertical velocity components and the pressure are computed on different grids. Also the different sets of equations are collocated in different points (see Figure 6.1): in particular, the horizontal equilibrium equations are collocated on the horizontal velocity points; the vertical equilibrium equations are collocated on the vertical velocity points, and the incompressibility constraints are collocated on the pressure points. One of the advantages of staggered grids is the fact that for Dirichlet boundary conditions (that is, where the fluid velocity is known) no boundary conditions for the pressure are required. Moreover, staggered grids preserve the properties of continous differential operators: as an example, the superimposition of the discrete divergence operator on the discrete gradient operator leads to a correct discretization of the Laplace operator. The main drawback for MAC grids is that they cannot be used for non-regular distributions of nodes, and thus they cannot be extended, in general, to meshless methods.

### 6.2 Stokes problem: alternative formulations

In the present section we introduce alternative formulations presented in the literature for the solution of the Stokes problem. Thanks to some modifications of the original Stokes equations (6.5), for such formulations the respect of an inf-sup condition is not required, and therefore neither special tricks for the discretization, nor special grids, are necessary, making such formulations well suited to be approximated through meshless methods.

### 6.2.1 Stokes equations in the Pressure Poisson form

In the first three formulations presented in this section the incompressibility constraint is replaced by a different equation, obtained applying the divergence operator to the equilibrium equation of System (6.5). Accordingly, we have

$$
\begin{equation*}
\nabla \cdot\left(\rho \frac{\partial \mathbf{u}}{\partial t}\right)=\nabla \cdot(-\nabla p+\mu \Delta \mathbf{u}+\mathbf{b}) \tag{6.6}
\end{equation*}
$$

that can be rewritten in the form

$$
\begin{equation*}
\Delta p-\mu \nabla \cdot(\Delta \mathbf{u})=\nabla \cdot \mathbf{b} \tag{6.7}
\end{equation*}
$$

and then further simplified exploiting the incompressibility constraint $\nabla \cdot \mathbf{u}=0$, obtaining

$$
\begin{equation*}
\Delta p=\nabla \cdot \mathbf{b} \tag{6.8}
\end{equation*}
$$

The constraint conditions (6.7) and (6.8) are both Poisson equation for the pressure, and therefore they are known as Pressure Poisson Equations. In the literature there has been a great discussion on whether boundary conditions are needed for equations (6.7) and (6.8). Sani et al. (2006) propose three different formulations:

1. In the first formulation (referred to, in the following, as formulation S1) the equilibrium equation is coupled to the constraint equation (6.7), obtaining the following set of equations

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{u}}{\partial t}+\nabla p=\mu \Delta \mathbf{u}+\mathbf{b}  \tag{6.9}\\
\Delta p-\mu \nabla \cdot(\Delta \mathbf{u})=\nabla \cdot \mathbf{b}
\end{array}\right.
$$

on which Dirichlet or Neumann boundary conditions are imposed on the equilibrium equations, and no boundary conditions are imposed on the constraint equation.
2. In the second formulation (referred to as S 2 ) the equilibrium equation is coupled to the constraint equation (6.8)

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{u}}{\partial t}+\nabla p=\mu \Delta \mathbf{u}+\mathbf{b}  \tag{6.10}\\
\Delta p=\nabla \cdot \mathbf{b}
\end{array}\right.
$$

The boundary conditions for the constraint equation, in this case, are the projections of the equilibrium equation on the boundary outward normal vector.

$$
\begin{equation*}
(\rho \partial \mathbf{u} / \partial t+\nabla p-\mu \Delta \mathbf{u}-\mathbf{b}) \cdot \mathbf{n}=0 \tag{6.11}
\end{equation*}
$$

3. In the third formulation (referred to as $S 3$ ) the equilibrium and the constraint equations are the same as Equations (6.10), but the boundary conditions for the constraint equation is the original incompressibility condition $\nabla \cdot \mathbf{u}=0$.

### 6.2.2 A pseudo-compressibility formulation of the Stokes problem

A commonly used formulation for the Stokes problem in primitive variables is

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{u}}{\partial t}+\nabla p=\mu \Delta \mathbf{u}+\mathbf{b}  \tag{6.12}\\
\nabla \cdot \mathbf{u}-\varepsilon \Delta p=0
\end{array}\right.
$$

where a relaxation term $\varepsilon \Delta p$ is introduced in the incompressibility condition. Such a formulation, discussed by Brezzi and Douglas Jr (1988) in the framework of Galerkin methods, belongs to the class of the pseudo-compressibility methods, since a perturbation is introduced in the continuity constraint. The addition of such a perturbation results in a smoothing of the pressure field, alleviating the effects of checkerboard instability. However, the parameter $\varepsilon$ has to be properly set: it has to be not excessively small, in order to have a regularizing effect on the pressure field; and it has to be not too high, since it introduces an error in the original incompressibility condition (Quarteroni et al., 2000).

The boundary conditions for the equilibrium equations are the usual conditions on velocity or stress. The boundary condition adopted for the constraint equation is the continuity equation $\nabla \cdot \mathbf{u}=0$. This formulation will be referred in the following as S 4 .

### 6.2.3 The gauge method

The gauge method (Wang and Liu, 2000; E and Liu, 2003) is a continous formulation of the Stokes problem based on the following change of variables

$$
\begin{equation*}
\mathbf{u}=\mathbf{a}-\nabla \phi \tag{6.13}
\end{equation*}
$$

in Equations (6.5). Accordingly the modified set of equations is

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{a}}{\partial t}-\rho \frac{\partial \nabla \phi}{\partial t}=-\nabla p+\mu \Delta \mathbf{a}-\mu \Delta \nabla \phi+\mathbf{b}  \tag{6.14}\\
\nabla \cdot \mathbf{a}=-\Delta \phi
\end{array}\right.
$$

that can be rewritten as

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{a}}{\partial t}=\mu \Delta \mathbf{a}+\mathbf{b}  \tag{6.15}\\
\nabla \cdot \mathbf{a}=-\Delta \phi
\end{array}\right.
$$

if the pressure $p$ is written as

$$
\begin{equation*}
p=\rho \frac{\partial \phi}{\partial t}-\mu \Delta \phi \tag{6.16}
\end{equation*}
$$

From Equation (6.15) we observe that the variables a and $\phi$ are decoupled in the linear equilibrium equation. However, in this formulation the equilibrium equation cannot be solved separately from the constraint equation, since a and $\phi$ are coupled in the boundary conditions, as described in the following

Dirichlet boundary conditions. The expressions of Dirichlet boundary conditions are

$$
\left\{\begin{array}{l}
\mathbf{u} \cdot \mathbf{n}=\bar{u}_{n}  \tag{6.17}\\
\mathbf{u} \cdot \mathbf{t}=\bar{u}_{\tau}
\end{array}\right.
$$

that can be rewritten, using Equation (6.13), in the form

$$
\left\{\begin{array}{l}
(\mathbf{a}+\nabla \phi) \cdot \mathbf{n}=\bar{u}_{n}  \tag{6.18}\\
(\mathbf{a}+\nabla \phi) \cdot \mathbf{t}=\bar{u}_{\tau}
\end{array}\right.
$$

E and Liu (2003) consider the case of homogeneous Dirichlet boundary conditions and propose two different possible choices:
Case 1: $\quad \frac{\partial \phi}{\partial \mathbf{n}}=0$
$\mathbf{a} \cdot \mathbf{n}=0$
$\mathbf{a} \cdot \mathbf{t}=\frac{\partial \phi}{\partial \mathbf{t}}$
Case 2: $\quad \phi=0$
$\mathbf{a} \cdot \mathbf{n}=\frac{\partial \phi}{\partial n}$
$\mathbf{a} \cdot \mathbf{t}=0$

For future discussion, we remark that conditions (6.19) and (6.20) are obtained, after some algebra, from Equation (6.18):

1. The first two boundary conditions of (6.19) are obtained splitting the first equation of
(6.18) in two parts, the first containing the variable a and the second containing the variable $\phi$, while the last boundary condition of (6.19) is simply the second equation of (6.18);
2. The set of boundary conditions (6.20), on the contrary, are obtained using a similar procedure, but splitting the tangential boundary condition of (6.18) instead of the normal boundary condition.

Neumann boundary conditions. The case of Neumann boundary conditions has not been tackled so far within the context of the gauge method, and therefore we extend to incorporate also the Neumann boundary conditions. The stress tensor $\sigma$ is expressed by the relation

$$
\begin{equation*}
\boldsymbol{\sigma}=-p \mathbf{I}+\mu\left(\nabla \mathbf{u}+\nabla \mathbf{u}^{T}\right) \tag{6.21}
\end{equation*}
$$

which, using Equation (6.13), can be rewritten as

$$
\begin{equation*}
\boldsymbol{\sigma}=-\left(\rho \frac{\partial \phi}{\partial t}-\mu \Delta \phi\right) \mathbf{I}+\mu\left(\nabla \mathbf{a}+\nabla \mathbf{a}^{T}-2 \nabla \nabla \phi\right) \tag{6.22}
\end{equation*}
$$

In Equation (6.22) the term $\nabla \nabla \phi$ is the second gradient of the scalar field $\phi$, i.e., , in index notation

$$
\begin{equation*}
(\nabla \nabla \phi)_{i j}=\frac{\partial^{2} \phi}{\partial x_{i} \partial x_{j}} \tag{6.23}
\end{equation*}
$$

Introducing the change of variables (6.13) in the expressions of the components of the outward stress at the boundary

$$
\left\{\begin{array}{l}
\sigma \mathbf{n} \cdot \mathbf{n}=\bar{\sigma}_{n}  \tag{6.24}\\
\sigma \mathbf{n} \cdot \mathbf{t}=\bar{\sigma}_{t}
\end{array}\right.
$$

where $\mathbf{n}$ and $\mathbf{t}$ are again the outward normal and tangential unit vectors at the boundary of the domain, we obtain

$$
\left\{\begin{array}{l}
\mu[\Delta \phi+2(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{n}]+\mu\left(\nabla \mathbf{a}+\nabla \mathbf{a}^{T}\right) \mathbf{n} \cdot \mathbf{n}=\bar{\sigma}_{n}  \tag{6.25}\\
2 \mu(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{t}+\mu\left(\nabla \mathbf{a}+\nabla \mathbf{a}^{T}\right) \mathbf{n} \cdot \mathbf{t}=\bar{\sigma}_{t}
\end{array}\right.
$$

Restricting to the stationary case (i.e., $\partial \phi / \partial t=0$ ) and following a procedure similar to the case of Dirichlet boundary conditions, from Equation (6.25) we can obtain two different sets of Neumann boundary conditions. The first one is obtained splitting the normal component of the boundary conditions (6.25) and reads

$$
\left\{\begin{array}{l}
\mu\left(\nabla \mathbf{a}+\nabla \mathbf{a}^{T}\right) \mathbf{n} \cdot \mathbf{n}=\bar{\sigma}_{n}  \tag{6.26}\\
2 \mu(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{t}+\mu\left(\nabla \mathbf{a}+\nabla \mathbf{a}^{T}\right) \mathbf{n} \cdot \mathbf{t}=\bar{\sigma}_{t} \\
\mu[\Delta \phi+2(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{n}]=0
\end{array}\right.
$$

while the second one is obtained splitting the tangential component of System (6.25) and it
reads

$$
\left\{\begin{array}{l}
\mu[\Delta \phi+2(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{n}]+\mu\left(\nabla \mathbf{a}+\nabla \mathbf{a}^{T}\right) \mathbf{n} \cdot \mathbf{n}=\bar{\sigma}_{n}  \tag{6.27}\\
\mu\left(\nabla \mathbf{a}+\nabla \mathbf{a}^{T}\right) \mathbf{n} \cdot \mathbf{t}=\bar{\sigma}_{t} \\
2 \mu(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{t}=0
\end{array}\right.
$$

For both cases of boundary conditions (6.26) and (6.27), the last equation plays the role of boundary condition for the incompressibility equation.

In the numerical tests, we refer to the gauge method in stationary form with Dirichlet boundary conditions (6.19) with the abbreviation $\operatorname{S} 5-\mathrm{D}$; to the gauge method with Neumann boundary conditions (6.26) with the abbreviation $\mathrm{S} 5-\mathrm{N}-\mathrm{a}$; the gauge method with Neumann boundary conditions (6.27) with the abbreviation S5-N-b.

### 6.2.4 Summary of the formulations applied in the numerical tests

In Table 6.1 we summarize the formulations introduced in the previous paragraphs and indicate, for each one, the equation used as incompressiblity constraint and the corresponding boundary condition. These formulations are then tested in the next section on some benchmark problems.

| Formulation | Reference paper | constraint equation | BC for the constraint equation |
| :---: | :---: | :---: | :---: |
| S1 | Sani et al. (2006) | $\Delta p-\mu \nabla \cdot(\Delta \mathbf{u})=\nabla \cdot \mathbf{b}$ | none |
| S2 | Sani et al. (2006) | $\Delta p=\nabla \cdot \mathbf{b}$ | $(\rho \partial \mathbf{u} / \partial t+\nabla p-\mu \Delta \mathbf{u}-\mathbf{b}) \cdot \mathbf{n}=0$ |
| S3 | Sani et al. (2006) | $\Delta p=\nabla \cdot \mathbf{b}$ | $\nabla \cdot \mathbf{u}=0$ |
| S4 | Brezzi and Douglas Jr $(1988)$ | $\nabla \cdot \mathbf{u}-\varepsilon \Delta p=0$ | $\nabla \cdot \mathbf{u}=0$ |
| S5-D | Wang and Liu $(2000)$ | $\Delta \phi=-\nabla \cdot \mathbf{a}$ | $\partial \phi / \partial \mathbf{n}=0$ |
| S5-N-a |  | $\Delta \phi=-\nabla \cdot \mathbf{a}$ | $1 / \operatorname{Re}[\Delta \phi+2(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{n}]=0$ |
| S5-N-b |  | $\Delta \phi=-\nabla \cdot \mathbf{a}$ | $1 / \operatorname{Re}[\Delta \phi+2(\nabla \nabla \phi \mathbf{n}) \cdot \mathbf{t}]=0$ |

Table 6.1: Formulations of the steady Stokes problem analyzed in the present paper

### 6.3 Application of the MFPM to steady Stokes problems

In the following we use the Modified Finite Particle Method to approximate the spatial derivatives appearing in the different formulations introduced in the previous section. In the present section we restrict our attention to the stationary case (that is, $\partial \mathbf{u} / \partial t=0$ ). We first solve the well-known benchmark of the lid-driven cavity flow and we focus on how formulations S1, S2, and S3 (where the incompressibility constraint is not enforced strongly) satisfy the incompressibility condition $\nabla \cdot \mathbf{u}=0$. We then apply formulations $\mathrm{S} 3, \mathrm{~S} 4$, and S 5 on a square with a polynomial exact solution, on a quarter of annulus with a polynomial solution, and on a square under a uniform body load.

### 6.3.1 The lid-driven cavity

The geometry of the lid-driven cavity is a square of side $L$, as depicted in Figure 6.2; we set $L=1 m, \mu=1 \mathrm{~kg} / \mathrm{ms}$ as dynamic viscosity. The left, lower and right side of the square have
velocity $\mathbf{u}=0 \mathrm{~m} / \mathrm{s}$; the top side has a tangential velocity $\bar{U}=1 \mathrm{~m} / \mathrm{s}$ and a normal velocity $\mathbf{u} \cdot \mathbf{n}=0 \mathrm{~m} / \mathrm{s}$


Figure 6.2: The lid-driven cavity: geometry and boundary conditions
We start exploring the solution of the problem using formulations S1, S2, and S3 and paying particular attention to the method capability of enforcing the incompressibility constraint. In Figures $6.3,6.4$, and 6.5 we show the computed values of $\nabla \cdot \mathbf{u}$ using formulations S1, S2, and S3. We notice that in Figures 6.3 and 6.4 the divergence is significantly greater than zero, whereas in Figure 6.5 the divergence is close to zero everywhere except than in the top corners, where there is a discontinuity in the boundary conditions. These results suggest us to abandon formulations S1 and S2 and to proceed only with formulation S3.

### 6.3.2 Square with polynomial exact solution

We now consider a problem defined on a square domain $[-1,1] \mathrm{x}[-1,1]$ and constructed starting from the following manufactured exact solution:

$$
\left\{\begin{array}{l}
u(x, y)=20 x y^{3}  \tag{6.28}\\
v(x, y)=5\left(x^{4}-y^{4}\right) \\
p(x, y)=\left(60 x^{2} y-20 y^{3}+C\right)
\end{array}\right.
$$

The problem is formulated imposing no body loads in the interior of the domain, and Dirichlet boundary conditions on the whole boundary, in accordance to the analytical solution (7.30). The viscosity is set as $\mu=1 \mathrm{~kg} / \mathrm{ms}$.

The problem is solved using formulations S3, S4, and S5. The relaxation parameter of formulation S 4 is set as $\varepsilon=10^{-4}$. The convergence diagrams of the error related to the velocity field is reported in Figure 6.6. In particular the gauge method (formulation S5-D) shows higher order convergence with respect to the expected second order, and formulation S4 shows an even higher accuracy in the left part, and a second-order accuracy at the right side of the diagram.


Figure 6.3: Lid-driven cavity: divergence of the velocity using formulation S1


Figure 6.4: Lid-driven cavity: divergence of the velocity using formulation S2


Figure 6.5: Lid-driven cavity: divergence of the velocity using formulation S3

In Figure 6.7 the error related to pressure field is shown. In this case, formulation S5-D is the only one which shows a constant slope (1.78) of the error curve, even if slightly below the expected second-order accuracy. Formulation S3 shows correct accuracy at the left side of the diagram, and a little flattening in the right side of the diagram, while formulation S4 shows high convergence in the left side of the diagram (higher than second-order) and a complete flattening of the curve in the right side. We remark that such a flattening can be ascribed to the relaxation term $\varepsilon \Delta p$ in the continuity equation of formulation S4, that, at the level of discretization reached on the right zone of Figure 6.7, introduces an error on the pressure field higher than the relaxation effects.

The same problem has been studied also imposing Neumann boundary conditions on the left and right sides of the domain and Dirichlet boundary conditions on the top and lower sides, and using gauge formulations $\mathrm{S} 5-\mathrm{N}-\mathrm{a}$ and $\mathrm{S} 5-\mathrm{N}-\mathrm{b}$ for the numerical solution. In the case of formulation S5-N-a, we discretized unknown fields in space using both second-order and third-order accurate MFPM discrete differential operators. From the convergence plots shown in Figure 6.8 we notice that in both cases the slope of the error curve is one order below the expected one. We argue that this is due to the fact that second derivatives are used in the boundary conditions. This is also confirmed using formulation $55-\mathrm{N}-\mathrm{b}$ and second order accurate MFPM differential operators. In fact, from Figure 6.8, the corresponding error curve shows first-order accuracy.


Figure 6.6: Square with exact solution (7.30): convergence diagram of the velocity error with formulations S3, S4 and S5-D


Figure 6.7: Square with exact solution (7.30): convergence diagram of the pressure error with formulations S3, S4 and S5-D


Figure 6.8: Problem with exact solution (7.30): convergence diagram of the error using formulations $\mathrm{S} 5-\mathrm{N}-\mathrm{a}$ with second and third order acurate differential operators, and formulation $\mathrm{S} 5-\mathrm{N}-\mathrm{b}$ using second order accurate differential operators. In all case the method experiences a loss of convergence.

### 6.3.3 Quarter of annulus under body load

We now consider a quarter of annulus, clamped on its entire boundary, under a polynomial body load. The geometry of the problem is depicted in Figure 7.7, with $R=4$ and $r=1$.

The analytical solution of the problem is set equal to

$$
\left\{\begin{array}{l}
u=10^{-6} x^{2} y^{4}\left(x^{2}+y^{2}-16\right)\left(x^{2}+y^{2}-1\right)\left(5 x^{4}+18 x^{2} y^{2}-85 x^{2}+13 y^{4}+80-153 y^{2}\right)  \tag{6.29}\\
v=-2 \cdot 10^{-6} x y^{5}\left(x^{2}+y^{2}-16\right)\left(x^{2}+y^{2}-1\right)\left(5 x^{4}-51 x^{2}+6 x^{2} y^{2}-17 y^{2}+16+y^{4}\right)
\end{array}\right.
$$

The internal body loads are obtained using the manufactured solution (7.33). The problem has been studied in Auricchio et al. (2007) using a stream function formulation and isogeometric analysis for the spatial discretization, exploiting the high regularity of isogeometric shape-functions, and also the possibility of reproducing exactly the geometry of the domain.

In the following we investigate how the selection algorithm of supporting particles for the derivative approximation influences the quality of the solution on each formulation. The test is relevant since, due to the particular geometry of this problem, a particle placed on the inner radius of the annulus has, as closest particles, other particles placed on only two quasiparallel lines, leading to an ill-conditioned matrix $\mathbf{A}^{i}$. In particular, on a regular distribution of particles (see Figure 6.9(b)), we test three different algorithms:

1. searching algorithm 1: the minimum number of supporting nodes for derivative approximation is $N_{i}=9$. The adopted threshold condition number for matrix $\mathbf{A}^{i}$ is


Figure 6.9: Quarter of annulus clamped on all its edges

$$
C_{\max }=4 \cdot 10^{8}
$$

2. searching algorithm 2: the minimum number of supporting nodes for derivative approximation is $N_{i}=15$. The threshold condition number for matrix $\mathbf{A}^{i}$ is $C_{\max }=4 \cdot 10^{8}$;
3. searching algorithm 3: nodes are selected exploiting the particular topology of the problem and the regularity of the particle distribution. In this case each particle can be marked using indices $i$ and $j$ on a cylindrical reference frame, and then, for a particle denoted with $(i, j)$, the supporting particles are the ones between $i-1$ and $i+1$, and those between $j-1$ and $j+1$. Afterwards, derivative approximations are obtained using the usual MFPM procedure. It is clear that this searching algorithm is not general, and can be used only for regular distributions: however we present it since it is the most accurate solution that the MFPM can achieve.

In Figures $6.10(\mathrm{a})$ and $6.11(\mathrm{a})$ we show the supporting nodes of a particle placed at $x=$ $y=\sqrt{2} / 2$ selected using searching algorithms 1 and 2 . In the first case, the particles included in the set of supporting nodes are 10 , obtaining a condition number $C=8.15110^{3}$ of the matrix $\mathbf{A}^{i}$. In the second case the supporting particles are not placed on only two parallel lines, and consequently the condition number is strongly reduced $\left(C=2.1810^{2}\right)$.

In Figure 6.12(a) we show supporting nodes obtained using searching algorithm 3. Particles are placed on three different lines, leading to better results in terms of accuracy and error convergence, as shown comparing Figures 6.10(b), 6.11(b), and 6.12(b).

For all three searching algorithms, we notice that the formulation S5-D exhibits always better performances with respect to the other formulations. Formulations S 3 shows always the worst performances in terms of magnitude of the error, even if a second-order accuracy is typically attained by all the investigated searching algorithms. Finally, it has to be noted that formulation S 4 shows some convergence oscillations, even if an average second-order accuracy

(a) Supporting particles of node $\mathbf{x}_{i}$

(b) Convergence diagram of the error using formulations S3, S4 and S5-D, with second-order accurate MFPM

Figure 6.10: Problem of a quarter of annulus with exact solution (7.33) solved using searching algorithm 1

(a) Supporting particles of node $\mathbf{x}_{i}$

(b) Convergence diagram of the error using formulations S3, S4 and S5-D, with second-order accurate MFPM

Figure 6.11: Problem of a quarter of annulus with exact solution (7.33) solved using searching algorithm

(a) Supporting particles of node $\mathbf{x}_{i}$

(b) Convergence diagram of the error using formulations S3, S4 and S5-D, with second-order accurate MFPM

Figure 6.12: Problem of a quarter of annulus with exact solution (7.33) solved using searching algorithm 3
is obtained. Moreover, the error magnitude of formulation S4 is often comparable with the one shown by formulation S5-D.

### 6.3.4 Square clamped on two edges under a vertical body load

The geometry of the following problem is a square in the domain of side $L$ under a vertical body load, as depicted in Figure 6.13. The problem is an incompressible solid mechanics application, and here we highlight that the governing equations do not change, but only the interpretation of variables is different with respect to the fluid case. We set $L=1, \mu=40 \mathrm{~kg} / \mathrm{ms}$, and $\mathbf{b}=\left[\begin{array}{ll}0 & -80\end{array}\right]^{T} N / m^{2}$. The imposed boundary conditions are

$$
\left\{\begin{array}{rll}
\mathbf{u}=\mathbf{0} & \text { left and lower sides }  \tag{6.30}\\
\boldsymbol{\sigma} \mathbf{n}= & \mathbf{0} & \text { right and top sides }
\end{array}\right.
$$

The present problem has been solved in Auricchio et al. (2007) using the stream-function formulation and an isogeometric approach for the spatial discretization.

The reference solutions are the vertical displacement in the point $A$, the horizontal and vertical displacements in the point $B$, and the horizontal displacement in the point $C$, computed with the commercial Finite Element code "ABAQUS" using an overkilled discretization, and we compute the relative error in some sampling points, as reported in Auricchio et al. (2007). The numerical solutions are computed using formulations S3, S4 and S5-N-a.The relaxation parameter of formulation S4 has been set as $\varepsilon=10^{-4}$. The convergence plots of the error are reported in Figures 6.14, 6.15, and 6.16.

We notice that formulations S3 and S4 exhibit the expected second-order convergence of the error, whereas formulation $\mathrm{S} 5-\mathrm{N}$-a experiences a loss of convergence order, similar to the


Figure 6.13: Square clamped on two edges under a vertical body load: geometry and boundary conditions


Figure 6.14: Square clamped on two edges: convergence diagram of the error using formulation S3


Figure 6.15: Square clamped on two edges: convergence diagram of the error using formulation S4


Figure 6.16: Square clamped on two edges: convergence diagram of the error using formulation S5-N-a
case of the previous problem with Neumann boundary conditions.

### 6.4 Non-stationary Stokes problem: continous formulations

Following what is usually done for the equations of linear elasticity, the most natural idea for the solution of a non-stationary Stokes problem is to solve jointly the equation of motion and the incompressibility constraint. This procedure is extremely costly, since at each time step an inversion of a 3 Nx 3 N sparse matrix is required, and time integration has to be necessarily done implicitly, since the constraint equation is not depending on time. Moreover, the joint solution of motion and incompressibility equations requires the spatial discretization to satisfy the inf-sup condition.

Therefore the procedure commonly adopted in the literature for the solution of nonstationary Stokes problems consists in enforcing equilibrium and incompressibility in two different substeps of each time integration step, reducing the total dimensions of the matrices to invert, and hence reducing the computational cost of the method. In the following we describe the Chorin algorithm, that was the first algorithm introducing this decoupling between substeps, and its developments.

### 6.4.1 The original Chorin algorithm

The Chorin algorithm (Chorin, 1967, 1968) consists in dividing each time-integration step in three substeps:

1. In the first substep, a guess velocity $\mathbf{u}^{*}$ is computed through the solution of a modified equilibrium equation, in which the pressure term is omitted; accordingly Equation (6.5) reduces to the following

$$
\left\{\begin{array}{l}
\rho \frac{\mathbf{u}^{*}-\mathbf{u}^{n}}{\Delta t}=\mu \Delta \mathbf{u}^{n}+\mathbf{b}  \tag{6.31}\\
+ \text { suitable boundary condition on } \mathbf{u}^{*}
\end{array}\right.
$$

The guess velocity $\mathbf{u}^{*}$ does not respect, in general, the incompressibility constraint;
2. In the second step the pressure $p^{n+1}$ is computed solving the system

$$
\left\{\begin{array}{l}
\rho \frac{\mathbf{u}^{\mathbf{n}+\mathbf{1}}-\mathbf{u}^{*}}{\Delta t}=-\nabla p^{n+1}  \tag{6.32}\\
\nabla \cdot \mathbf{u}^{n+1}=0 \\
+ \text { suitable boundary conditions on } p^{n+1}
\end{array}\right.
$$

where the incompressibility constraint at the time step $n+1$ is enforced. Equations (6.32) can be rewritten in the Poisson form as follows

$$
\left\{\begin{array}{l}
\Delta t \Delta p^{n+1}=\nabla \cdot \mathbf{u}^{*}  \tag{6.33}\\
+ \text { suitable boundary conditions on } p^{n+1}
\end{array}\right.
$$

3. Finally, the velocity $\mathbf{u}^{n+1}$ is corrected using the guess solution $\mathbf{u}^{*}$ and $p^{n+1}$

$$
\begin{equation*}
\mathbf{u}^{n+1}=\mathbf{u}^{*}-\frac{1}{\rho} \nabla p^{n+1} \tag{6.34}
\end{equation*}
$$

We remark that the sum of Equations (6.31) and (6.32) gives a correct time discretization of the Stokes problem, with an explicit time approximation for the velocity and an implicit approximation for the pressure.

### 6.4.2 Developments of the Chorin algorithm

Following the pioneering work by Chorin, many other algorithms have been introduced, especially with the purpose of improving the accuracy near the boundary. In order to reach this goal, one may act on the boundary conditions to be imposed on Equation (6.31), the boundary conditions to be imposed on Equation (6.32), on the pressure update (6.34).

A deep analysis of the Chorin algorithm is proposed in E and Liu (1995), where the choice of the proper boundary conditions to be imposed at the different steps of the algorithm is discussed, in order to reduce the error introduced in the interior of the domain when inaccurate (or wrong) boundary conditions are imposed.

Furthermore Brown et al. (2001) accurately review a wide number of algorithms, discussing the convergence order in time, that is a consequence of the choice of boundary conditions at the different algorithm substeps. All the discussed algorithms are presented in the form

$$
\begin{gather*}
\left\{\begin{array}{l}
\rho \frac{\mathbf{u}^{*}-\mathbf{u}^{n}}{\Delta t}=-\nabla q+\frac{\mu}{2} \Delta\left(\mathbf{u}^{*}+\mathbf{u}^{n}\right) \\
B\left(\mathbf{u}^{*}\right)=0 \quad x \in \partial \Omega
\end{array}\right.  \tag{6.35a}\\
\left\{\begin{array}{l}
\mathbf{u}^{n+1}=\mathbf{u}^{*}-\Delta t \nabla \phi^{n+1} \\
\nabla \cdot \mathbf{u}^{n+1}=0 \\
B C\left(\phi^{n+1}\right)=0 \quad x \in \partial \Omega
\end{array}\right.  \tag{6.35b}\\
p^{n+1 / 2}=q+L\left(\phi^{n+1}\right) \tag{6.35c}
\end{gather*}
$$

where $q, \phi$ are auxiliary variables related to the pressure, $L$ is a linear operator, and $B$ and $B C$ are suitable boundary conditions to be imposed on $\mathbf{u}^{*}$ and $\phi$ respectively. We remark the fact that in (6.35) a second-order implicit scheme has been chosen for the time advance.

In the following we focus on three algorithms described in Brown et al. (2001), which follow the substeps $(6.35 \mathrm{a}),(6.35 \mathrm{~b})$, and $(6.35 \mathrm{c})$, characterized by different choices of $q$ and $L$ :

1. The first algorithm was introduced by Bell et al. (1989) and is characterized by the following choice: $q=p^{n+1 / 2}$ and $L=I$. In the present paper we refer to this algorithm with the abbreviation D1.
2. The second algorithm was introduced by Kim and Moin (1985) and is characterized by the following choice: $q=0$ and $L=I-\nu \Delta t / 2 \Delta$. In the present paper we refer to this algorithm with the abbreviation D2.
3. The third algorithm was introduced by Brown et al. (2001) and is characterized by the following choice: $q=0$ and $L=I-\mu \Delta t / 2 \Delta$. In the present paper we refer to this algorithm with the abbreviation D3.

For all the above formulations the selected boundary conditions are $B\left(\mathbf{u}^{*}\right)=\mathbf{u}^{*}-\mathbf{u}^{n+1}$ and $B C\left(\phi^{n+1}\right)=\partial \phi / \partial \mathbf{n}$.

We summarize the described algorithm in Table 6.2.

| Reference abbreviation | Reference paper | Linear operators |
| :---: | :---: | :---: |
| D1 | Bell et al. (1989) | $q=p^{n-1 / 2}, L=I$ |
| D2 | Kim and Moin (1985) | $q=p^{n-1 / 2}, L=I-\mu \Delta t / 2 \Delta$ |
| D3 | Brown et al. (2001) | $q=0, L=I-\mu \Delta t / 2 \Delta$ |

Table 6.2: Algorithms analyzed in the present paper for the non-stationary Stokes equations

### 6.4.3 Gauge method in dynamics

The non-stationary form of the gauge method proposed in E and Liu (2003) allows to solve the non-stationary Stokes problems in an efficient way using, at each time step, two different substeps instead of three:

1. In the first substep the equilibrium equation is advanced in time using any time integration scheme

$$
\begin{cases}\rho \frac{\partial \mathbf{a}}{\partial t}=\mu \Delta \mathbf{a}+\mathbf{b} & \mathbf{x} \in \Omega  \tag{6.36}\\ \mathbf{a}^{n+1} \cdot \mathbf{n}=0 & \\ \mathbf{a}^{n+1} \cdot \mathbf{t}=-\frac{\partial \phi^{n}}{\partial \mathbf{t}} & \mathbf{x} \in \Gamma\end{cases}
$$

Differently from the Chorin algorithm, in this case the field a has not to be corrected.
2. In the second time step the gauge variable $\phi^{n+1}$ is computed through the solution of the following Poisson problem

$$
\left\{\begin{array}{l}
\Delta \phi^{n+1}=-\nabla \cdot \mathbf{a}^{n+1} \quad \mathbf{x} \in \Omega  \tag{6.37}\\
\frac{\partial \phi^{n+1}}{\partial \mathbf{n}}=0 \quad \mathbf{x} \in \Gamma
\end{array}\right.
$$

We finally remark that the tangential boundary condition of substep 1 takes into account the value of the gauge variable $\phi$ at the time step $n$ rather than at the time step $n+1$, allowing to decouple the solution of substep 1 from the solution of substep 2.

In the following section we refer to the non-stationary gauge method as formulation D4.

### 6.5 Application of the MFPM to a non-stationary Stokes problems

In the present section we apply the MFPM spatial discretization to formulations from D1 to D4. The test case is a square in the domain $[-1,1] \mathrm{x}[-1,1]$ with exact solution

$$
\left\{\begin{array}{l}
u(x, y, t)=20 x y^{3} \sin (2 \pi t)  \tag{6.38}\\
v(x, y, t)=5\left(x^{4}-y^{4}\right) \sin (2 \pi t) \\
p(x, y, t)=\left(60 x^{2} y-20 y^{3}\right) \sin (2 \pi t)
\end{array}\right.
$$

The problem is governed by the non-stationary Stokes equations (with material properties $\rho=1 \mathrm{~kg} / \mathrm{m}^{3}$ and $\mu=1 \mathrm{~kg} / \mathrm{ms}$ ) subjected to homogeneous initial conditions and Dirichlet boundary conditions in accordance with Equation (6.38) on the whole boundary.

### 6.5.1 Solution using algorithms D1, D2 and D3

In Figures $6.17,6.18$, and 6.19 we show the convergence diagrams of the error obtained applying MFPM spatial discretization on formulations D1, D2, and D3. For each formulation, the analyses have been run using different time steps. The 2-norm error is computed at $t=0.25$


Figure 6.17: Problem with exact solution (6.38): convergence diagram of the error in space with formulation D1

Using formulation D1 (Figure 6.17) we notice the expected second-order rate of convergence of the error for $\Delta t=10^{-3}$ and $\Delta t=10^{-4}$, while for $\Delta t=10^{-2}$ the error in time is dominant, worsening the convergence order of the error in space. Using formulation D2 (Figure 6.18)


Figure 6.18: Problem with exact solution (6.38): convergence diagram of the error in space with formulation D2


Figure 6.19: Problem with exact solution (6.38): convergence diagram of the error in space with formulation D3
we notice that the convergence rate is still optimal for $\Delta t=10^{-3}$ and $\Delta t=10^{-4}$, and we notice also an appreciable improvement of the solution with $\Delta t=10^{-2}$. Using formulation D3 (Figure 6.19) we notice that the convergence rate in space is lost for all $\Delta t$.

### 6.5.2 Solution using algorithm D4

The problem with exact solution (6.38) is also solved using formulation D 4 (transient gauge method) with three different time integration schemes for the equilibrium equation of System (6.36):

1. an implicit Euler scheme (first order accurate in time). The equilibrium equation is discretized as follows:

$$
\begin{equation*}
\frac{\mathbf{a}^{n+1}-\mathbf{a}^{n}}{\Delta t}=\Delta \mathbf{a}^{n+1}+\mathbf{b}^{n+1} \tag{6.39}
\end{equation*}
$$

2. a Cranck-Nicholson scheme (implicit scheme, second order accurate in time). The equilibrium equation is discretized as follows:

$$
\begin{equation*}
\frac{\mathbf{a}^{n+1}-\mathbf{a}^{n}}{\Delta t}=\frac{1}{2}\left(\Delta \mathbf{a}^{n+1}+\Delta \mathbf{a}^{n}\right)+\frac{1}{2}\left(\mathbf{b}^{n+1}+\mathbf{b}^{n}\right) \tag{6.40}
\end{equation*}
$$

3. an explicit fourth order accurate Runge-Kutta time integration scheme

In the case of explicit Euler time discretization (6.39) the convergence diagram of the error in space is shown in Figure 6.20. We notice that only using $\Delta t=10^{-4}$ there is a correct second-order accuracy of the solution, while using the other time steps the expected accuracy is lost due to the predominance of the error in time.

In the case of Cranck-Nicholson time discretization (6.40) we obtain the convergence diagram of the error in space reported in Figure 6.21, from which we notice that the error in time is small enough not to affect the second-order accuracy of the solution in space.

The convergence diagram of the error obtained using the fourth-order accurate RungeKutta scheme is presented in Figure 6.22. We notice that for time steps $\Delta t=10^{-2}$ and $\Delta t=$ $10^{-3}$, the scheme experiences numerical instability for more accurate space discretizations, due to the stability limits of the Runge-Kutta scheme.


Figure 6.20: Problem with exact solution (6.38): convergence diagram of the error in space with formulation D4 and scheme (6.39) for the advance in time


Figure 6.21: Problem with exact solution (6.38): convergence diagram of the error in space with formulation D4 and scheme (6.40) for the advance in time


Figure 6.22: Problem with exact solution (6.38): convergence diagram of the error in space with formulation D4, using a fourth-order explicit Runge-Kutta scheme for the advance in time

## Chapter 7

## Modified Finite Particle Method in the framework of the Least Square Residual Method

The application of the Modified Finite Particle Method in its novel formulation (see chapter 3 for details) to full incompressible bodies implies, as shown in Chapter 6, that the equations governing the behaviour of full incompressible bodies have to be reformulated, in order to overcome the need of respecting the inf-sup condition.

In our research we would like to solve the original equations of incompressibility, without recurring to any trick or modification of the governing equations. This is why here we introduce an extended formulation of the MFPM, that can be used for the discretization of an incompressibility problem, using a Least Square Residual Method for the solution of the final problem. In this chapter therefore we show the extended MFPM formulation and then we discretize the Stokes equations using the MFPM in conjunction with the Least Square Residual Method, following the idea of Chi et al. (2014) for the case of Radial Basis collocation, and show the solution of some benchmarks. Finally we extend the formulation also to non-linear problems, and solve the famous benchmark of the lid-driven cavity.

### 7.1 Modified Finite Particle Method: the extended formulation

The extended formulation of the Modified Finite Particle Method is slightly different from the version published in Asprone et al. (2014). The present version considers two different sets of points for the approximation:

1. the collocation points, indicated in the following as $\mathbf{x}=\left[\begin{array}{lll}x & y & z\end{array}\right]^{T}$, are the points where functions and derivatives are computed. This node distribution is placed within the physical domain of the problem under consideration, and is the node set on which equations are collocated. The total number of collocation points is indicated with $N_{C}$;
2. the control nodes, indicated in the following as $\boldsymbol{\xi}=\left[\begin{array}{lll}\xi & \eta & \zeta\end{array}\right]^{T}$, are the nodes where we place the degrees of freedom in terms of which we express functions and derivatives.

Control nodes do not have immediate physical evidence and hence they can be placed in any convenient way in the domain, i.e., on a Cartesian, equispaced grid. We remark that this choice does not affect the characteristic of the MFPM of being a meshless method, since collocation nodes can assume any position, even extremely unstructured, within the physical domain. The total number of control nodes used for the approximation is indicated with $N_{S}$.

The first step of the approximation procedure of a scalar function $u(\mathbf{x})$ and its spatial derivatives is the computation of the Taylor series expansion of $u(\mathbf{x})$, centered in a collocation point $\mathbf{x}_{i}$ and expanded up to the second order

$$
\begin{align*}
u(\boldsymbol{\xi})=u\left(\mathbf{x}_{i}\right) & +D_{x} u\left(\mathbf{x}_{i}\right)\left(\xi-x_{i}\right)+D_{y} u\left(\mathbf{x}_{i}\right)\left(\eta-y_{i}\right)+D_{z} u\left(\mathbf{x}_{i}\right)\left(\zeta-z_{i}\right)+ \\
& +\frac{1}{2} D_{x x}^{2} u\left(\mathbf{x}_{i}\right)\left(\xi-x_{i}\right)^{2}+\frac{1}{2} D_{y y}^{2} u\left(\mathbf{x}_{i}\right)\left(\eta-y_{i}\right)^{2}+\frac{1}{2} D_{z z}^{2} u\left(\mathbf{x}_{i}\right)\left(\zeta-z_{i}\right)^{2}+  \tag{7.1}\\
& +D_{x y}^{2} u\left(\mathbf{x}_{i}\right)\left(\xi-x_{i}\right)\left(\eta-y_{i}\right)+D_{y z}^{2} u\left(\mathbf{x}_{i}\right)\left(\eta-y_{i}\right)\left(\zeta-z_{i}\right)+ \\
& +D_{x z}^{2} u\left(\mathbf{x}_{i}\right)\left(\xi-x_{i}\right)\left(\zeta-z_{i}\right)
\end{align*}
$$

In a first stage we assume to know the nodal values of $u$ in the control nodes $\boldsymbol{\xi}$ : therefore Equation (7.1) contains 10 unknown terms (function and derivative values in the collocation point $\mathbf{x}_{i}$ ) and hence 10 equations are needed to compute their value. Therefore, for each collocation point $\mathbf{x}_{i}$ we select a subset $X_{i}$ of control nodes $\boldsymbol{\xi}_{j}$ which serve as auxiliary nodes for function and derivatives in $\mathbf{x}_{i}$. Then we evaluate Equation (7.1) in each node $\boldsymbol{\xi}_{j} \in X_{i}$ obtaining

$$
\begin{align*}
u\left(\boldsymbol{\xi}_{j}\right)=u\left(\mathbf{x}_{i}\right) & +D_{x} u\left(\mathbf{x}_{i}\right)\left(\xi_{j}-x_{i}\right)+D_{y} u\left(\mathbf{x}_{i}\right)\left(\eta_{j}-y_{i}\right)+D_{z} u\left(\mathbf{x}_{i}\right)\left(\zeta_{j}-z_{i}\right)+ \\
& +\frac{1}{2} D_{x x}^{2} u\left(\mathbf{x}_{i}\right)\left(\xi_{j}-x_{i}\right)^{2}+\frac{1}{2} D_{y y}^{2} u\left(\mathbf{x}_{i}\right)\left(\eta_{j}-y_{i}\right)^{2}+\frac{1}{2} D_{z z}^{2} u\left(\mathbf{x}_{i}\right)\left(\zeta_{j}-z_{i}\right)^{2}+  \tag{7.2}\\
& +D_{x y}^{2} u\left(\mathbf{x}_{i}\right)\left(\xi_{j}-x_{i}\right)\left(\eta_{j}-y_{i}\right)+D_{y z}^{2} u\left(\mathbf{x}_{i}\right)\left(\eta_{j}-y_{i}\right)\left(\zeta_{j}-z_{i}\right)+ \\
& +D_{x z}^{2} u\left(\mathbf{x}_{i}\right)\left(\xi_{j}-x_{i}\right)\left(\zeta_{j}-z_{i}\right)
\end{align*}
$$

We also introduce 10 known functions $W_{\alpha}^{i}=W_{\alpha}\left(\boldsymbol{\xi}-\mathbf{x}_{i}\right)$, evaluate them in the points $\boldsymbol{\xi}_{j} \in X_{i}$, and multiply the evaluations of the left- and right-hand sides of Equation (7.2) by the evaluations $W_{\alpha}^{i j}=W_{\alpha}\left(\boldsymbol{\xi}_{j}-\mathbf{x}_{i}\right)$. Finally we sum all products and obtain 10 equations of
the type

$$
\begin{align*}
u\left(\mathbf{x}_{i}\right) \sum_{j} W_{\alpha}^{i j} & +D_{x} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\xi_{j}-x_{i}\right) W_{\alpha}^{i j}+D_{y} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\eta_{j}-y_{i}\right) W_{\alpha}^{i j}+D_{z} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\zeta_{j}-z_{i}\right) W_{\alpha}^{i j}+ \\
& +\frac{1}{2} D_{x x}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\xi_{j}-x_{i}\right)^{2} W_{\alpha}^{i j}+\frac{1}{2} D_{y y}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\eta_{j}-y_{i}\right)^{2} W_{\alpha}^{i j}+ \\
& +\frac{1}{2} D_{z z}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\zeta_{j}-z_{i}\right)^{2} W_{\alpha}^{i j}+D_{x y}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\xi_{j}-x_{i}\right)\left(\eta_{j}-y_{i}\right) W_{\alpha}^{i j}+ \\
& +D_{y z}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\eta_{j}-y_{i}\right)\left(\zeta_{j}-z_{i}\right) W_{\alpha}^{i j}+D_{x z}^{2} u\left(\mathbf{x}_{i}\right) \sum_{j}\left(\xi_{j}-x_{i}\right)\left(\zeta_{j}-z_{i}\right) W_{\alpha}^{i j}= \\
& =\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{\alpha}^{i j} \quad \alpha=1, \ldots, 10 \tag{7.3}
\end{align*}
$$

that can be rearranged in matrix form as follows:

$$
\mathbf{A}^{i}\left(\begin{array}{c}
u\left(\mathbf{x}_{i}\right)  \tag{7.4}\\
D_{x} u\left(\mathbf{x}_{i}\right) \\
D_{y} u\left(\mathbf{x}_{i}\right) \\
D_{z} u\left(\mathbf{x}_{i}\right) \\
D_{x x}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{y y}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{z z}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{x y}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{y z}^{2} u\left(\mathbf{x}_{i}\right) \\
D_{x z}^{2} u\left(\mathbf{x}_{i}\right)
\end{array}\right)=\left(\begin{array}{c}
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{1}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{2}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{3}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{4}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{5}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{6}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{7}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{8}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{9}^{i j} \\
\sum_{j} u\left(\boldsymbol{\xi}_{j}\right) W_{10}^{i j}
\end{array}\right)
$$

The projection functions used in the applications in the present paper are

$$
\begin{array}{cc}
W_{1}^{i}=1 & W_{6}^{i}=\left(\eta-y_{i}\right)^{2} \\
W_{2}^{i}=\xi-x_{i} & W_{7}^{i}=\left(\zeta-z_{i}\right)^{2} \\
W_{3}^{i}=\eta-y_{i} & W_{8}^{i}=\left(\xi-x_{i}\right)\left(\eta-y_{i}\right) \\
W_{4}^{i}=\zeta-z_{i} & W_{9}^{i}=\left(\eta-y_{i}\right)\left(\zeta-z_{i}\right) \\
W_{5}^{i}=\left(\xi-x_{i}\right)^{2} & W_{10}^{i}=\left(\zeta-z_{i}\right)\left(\xi-x_{i}\right)
\end{array}
$$

Equation (7.4) is then rewritten in the form

$$
\begin{equation*}
\mathbf{A}^{i} \mathbf{D}\left(u_{i}\right)=\mathbb{W}^{i} \mathbf{u} \tag{7.5}
\end{equation*}
$$

where

$$
\mathbf{W}^{i}=\left[\begin{array}{lllll|l}
\mathbf{W}^{i 1} & \mid & \mathbf{W}^{i 2} & \mid & \ldots & \mathbf{W}^{i N_{i}} \tag{7.6}
\end{array}\right]
$$

and

$$
\mathbf{W}^{i j}=\left[\begin{array}{lllllll}
W_{1}^{i j} & \mid & W_{2}^{i j} & \mid & \ldots & \mid & W_{10}^{i j} \tag{7.7}
\end{array}\right]^{T}
$$

The vector $\mathbf{u}$ collects the known nodal values in the node set $\boldsymbol{\xi} \in X$. Then, by inverting (7.5), we obtain

$$
\begin{equation*}
\mathbf{D}\left(u_{i}\right)=\mathbf{E}^{i} \mathbb{W}^{i} \mathbf{u} \tag{7.8}
\end{equation*}
$$

where $\mathbf{E}^{i}=\left(\mathbf{A}^{i}\right)^{-1}$, and finally

$$
\begin{equation*}
\mathbf{D}\left(u_{i}\right)=\mathbb{D}^{i} \mathbf{u} \tag{7.9}
\end{equation*}
$$

with

$$
\begin{equation*}
\mathbb{D}^{i}=\mathbf{E}^{i} \mathbb{W}^{i} \tag{7.10}
\end{equation*}
$$

The $10 \times N_{i}$ operator $\mathbb{D}^{i}$, applied to $\mathbf{u}$, returns a 10 x 1 vector collecting all the approximations of functions and derivatives of $u(\mathbf{x})$ in the collocation point $\mathbf{x}_{i}$.

However, here we are interested in building 10 linear operators (II, $\mathbf{D}_{x}, \mathbf{D}_{y}, \mathbf{D}_{z}, \mathbf{D}_{x x}, \mathbf{D}_{y y}$, $\left.\mathbf{D}_{z z}, \mathbf{D}_{x y}, \mathbf{D}_{y z}, \mathbf{D}_{z x}\right)$ that, applied to the vector $\mathbf{u}$, return the evaluations of function and derivatives in all collocation points $\mathbf{x}$. These operators are simply built collecting, for each $i$, the correct row of $\mathbb{D}^{i}$, identified through Equation (7.4). For example, in order to build the linear operator $\mathbf{D}_{x}$ (the discrete counterpart of $\partial / \partial x$ ), we simply consider, for each $i$, the 2 nd row of $\mathbb{D}^{i}$. The final form of $\mathbf{D}_{x}$ is then

$$
\mathbf{D}_{x}=\left[\begin{array}{c}
\mathrm{D}_{2}^{1}  \tag{7.11}\\
\mathrm{D}_{2}^{2} \\
\cdots \\
\mathrm{D}_{2}^{N_{C}}
\end{array}\right]
$$

where $\mathbb{D}_{\alpha}^{i}$ is the $\alpha$-th row of $\mathbb{D}^{i}$.
Similarly, in order to retrieve the evaluations of $u(\mathbf{x})$ in the collocation points, we build the operator II such that

$$
\begin{equation*}
\left[u\left(\mathbf{x}_{i}\right)\right]_{i=1, \ldots, N_{C}}=\mathbb{I}\left[u\left(\boldsymbol{\xi}_{j}\right)\right]_{j=1, \ldots, N_{S}} \tag{7.12}
\end{equation*}
$$

whose rows are found selecting, for each $i$, the first row of $\mathbb{D}^{i}$ :

$$
\mathbb{I}=\left[\begin{array}{c}
\mathbb{D}_{1}^{1}  \tag{7.13}\\
\mathbb{D}_{1}^{2} \\
\cdots \\
\mathbb{D}_{1}^{N_{C}}
\end{array}\right]
$$

The columns of II can be interpreted as the evaluations of the $N_{S}$ shape functions in the collocation points.

### 7.2 Governing equations for incompressible flows

The governing equations of incompressible fluid flows are the well known Navier-Stokes equations

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{u}}{\partial t}+\rho \mathbf{u} \cdot \nabla \mathbf{u}=-\nabla p+\mu \Delta \mathbf{u}+\mathbf{b}  \tag{7.14}\\
\nabla \cdot \mathbf{u}=0
\end{array}\right.
$$

where the first equation expresses the conservation of the linear momentum, and the second equation expresses the incompressibility constraint. The variable $\rho$ is the fluid density, $\mathbf{u}$ is the velocity field, $p$ is the pressure field, $\mu$ is the dynamic viscosity, $\mathbf{b}$ is the vector of internal loads.

The Navier-Stokes equations are non-linear, due to the presence of the convective term $\rho \mathbf{u} \cdot \nabla \mathbf{u}$. Nevertheless, when viscous forces dominate inertial forces, such equations can be rewritten neglecting the convective term, obtaining

$$
\left\{\begin{array}{l}
\rho \frac{\partial \mathbf{u}}{\partial t}=-\nabla p+\mu \Delta \mathbf{u}+\mathbf{b}  \tag{7.15}\\
\nabla \cdot \mathbf{u}=0
\end{array}\right.
$$

known as Stokes equations. Both Systems (7.14) and (7.15) have to be completed with suitable boundary conditions, concerning the boundary velocity or the boundary outward stress.

In the present work we restrict to stationary flows, that is, $\partial \mathbf{u} / \partial t=0$; moreover, the work is divided in two parts: in the first part we concentrate on Stokes equations, in order to study how the Modified Finite Particle Method, in combination with the Least Square Residual Method, deals with the numerical limitation of the inf-sup condition; in the second part we focus on the solution of the complete Navier-Stokes equations, and show a numerical procedure to handle the non linearity.

### 7.2.1 Solution of the Stokes equations using the Modified Finite Particle Method and the Least Square Residual Method

In the spirit of collocation methods we discretize the steady Stokes equations using the Modified Finite Particle Method. The discrete linear system of equations is written in the form

$$
\left[\begin{array}{c}
\mathbf{K}_{e q}  \tag{7.16}\\
\mathbf{K}_{i n c} \\
\mathbf{K}_{\text {dir }} \\
\mathbf{K}_{\text {neum }}
\end{array}\right]\left(\begin{array}{c}
\hat{\mathbf{u}} \\
\hat{\mathbf{v}} \\
\hat{\mathbf{p}}
\end{array}\right)=\left(\begin{array}{c}
\hat{\mathbf{f}} \\
\mathbf{0} \\
\overline{\mathbf{u}} \\
\overline{\mathbf{t}}
\end{array}\right)
$$

$$
\begin{align*}
& \text { where } \\
& \mathbf{K}_{e q}=\left[\begin{array}{ccc}
\mu \mathbf{L} & \mathbf{0} & -\mathbf{D}_{x} \\
\mathbf{0} & \mu \mathbf{L} & -\mathbf{D}_{y}
\end{array}\right]  \tag{7.17}\\
& \mathbf{K}_{i n c}=\left[\begin{array}{lll}
\mathbf{D}_{x} & \mathbf{D}_{y} & \mathbf{0}
\end{array}\right]  \tag{7.18}\\
& \mathbf{K}_{d i r}=\left[\begin{array}{ccc}
\mathbf{n}_{x} \mathbb{I} & \mathbf{n}_{y} \mathbb{I} & 0 \\
\mathbf{t}_{x} \mathbb{I} & \mathbf{t}_{y} \mathbb{I} & \mathbf{0}
\end{array}\right]  \tag{7.19}\\
& \mathbf{K}_{\text {neum }}=\left[\begin{array}{ccc}
2 \mu\left(\mathbf{n}_{x}^{2} \mathbf{D}_{x}+\mathbf{n}_{x} \mathbf{n}_{y} \mathbf{D}_{y}\right) & 2 \mu\left(\mathbf{n}_{x} \mathbf{n}_{y} \mathbf{D}_{x}\right)+\mathbf{n}_{y}^{2} \mathbf{D}_{y} & -\mathbb{I} \\
2 \mu \mathbf{n}_{x} \mathbf{t}_{x} \mathbf{D}_{x}+\mu\left(\mathbf{n}_{x} \mathbf{t}_{y}+\mathbf{n}_{y} \mathbf{t}_{x}\right) \mathbf{D}_{y} & \mu\left(\mathbf{n}_{x} \mathbf{t}_{y}+\mathbf{n}_{y} \mathbf{t}_{x}\right) \mathbf{D}_{x}+2 \mu \mathbf{n}_{y} \mathbf{t}_{y} \mathbf{D}_{y} & \mathbf{0}
\end{array}\right] \tag{7.20}
\end{align*}
$$

In Equations (7.17), (7.18), (7.19), and (7.20) the matrix $\mathbf{L}=\mathbf{D}_{x x}+\mathbf{D}_{y y}$ is the discrete

Laplace operator, $\hat{\mathbf{u}}, \hat{\mathbf{v}}$, and $\hat{\mathbf{p}}$ are the nodal unknowns associated to the velocity components $u$ and $v$ and to the pressure $p, \hat{\mathbf{f}}$ is the vector of the internal loads at collocation points, $\overline{\mathbf{u}}$ is the vector of the known boundary displacements, and $\overline{\mathrm{t}}$ is the vector of the known boundary outward stress at Neumann collocation points. Finally $\mathbf{n}_{x}$ and $\mathbf{n}_{y}$ are square diagonal matrices collecting the values of the components of the boundary outward normal vector along the $x$ and $y$-direction; at the same way $\mathbf{t}_{x}$ and $\mathbf{t}_{y}$ are square diagonal matrices collecting the values of the boundary outward tangential vector along the $x$ - and $y$-direction.

When collocation and control points coincide, the values of the control unknowns can be retrieved by inverting system (7.16). Unfortunately the pressure field obtained through direct inversion of Equation (7.16) shows unphysical oscillations, known in the literature as pressure checkerboard instability.

In order to overcome such a numerical difficulty, here we use the Least Square Residual Method, following what has been successfully applied by Chi et al. (2014) using Radial Basis Functions for spatial discretization. The procedure consists in discretizing system (7.15) considering a number $N_{C}$ of collocation points higher than the number $N_{S}$ of control nodes; system (7.16) is therefore a rectangular, overdetermined system of algebraic equations, whose solution can be approximated through minimization of a squared error. Such an error is defined as

$$
\begin{equation*}
E=\|e\|^{2}=(\mathbf{K d}-\mathbf{f})^{T}(\mathbf{K d}-\mathbf{f}) \tag{7.21}
\end{equation*}
$$

where $\mathbf{K}$ is the stiffness matrix of Equation (7.16) and $\mathbf{d}=\left[\begin{array}{lll}\hat{\mathbf{u}} & \hat{\mathbf{v}} & \hat{\mathbf{p}}\end{array}\right]^{T}$. Error (7.21) can be furthermore expanded as

$$
\begin{align*}
E & =\left(\mathbf{K}_{e q} \mathbf{d}-\mathbf{f}\right)^{T}\left(\mathbf{K}_{e q} \mathbf{d}-\mathbf{f}\right)+\left(\mathbf{K}_{\text {inc }} \mathbf{d}\right)^{T}\left(\mathbf{K}_{\text {inc }} \mathbf{d}\right)+ \\
& +\left(\mathbf{K}_{\text {dir }} \mathbf{d}-\overline{\mathbf{u}}\right)^{T}\left(\mathbf{K}_{\text {dir }} \mathbf{d}-\overline{\mathbf{u}}\right)+\left(\mathbf{K}_{\text {neum }} \mathbf{d}-\overline{\mathbf{t}}\right)^{T}\left(\mathbf{K}_{\text {neum }} \mathbf{d}-\overline{\mathbf{t}}\right) \tag{7.22}
\end{align*}
$$

In Chi et al. (2014) it is noted that error (7.22) is unbalanced among its components. Therefore such error components are properly weighted, leading to the definition of a weighted error

$$
\begin{align*}
E_{w} & =\left(\mathbf{K}_{e q} \mathbf{d}-\mathbf{f}\right)^{T}\left(\mathbf{K}_{e q} \mathbf{d}-\mathbf{f}\right)+\left(\mathbf{K}_{\text {inc }} \mathbf{d}\right)^{T} \mathbb{A}_{\text {inc }}\left(\mathbf{K}_{\text {inc }} \mathbf{d}\right)+ \\
& +\left(\mathbf{K}_{\text {dir }} \mathbf{d}-\overline{\mathbf{u}}\right)^{T} \mathbb{A}_{\text {dir }}\left(\mathbf{K}_{\text {dir }} \mathbf{d}-\overline{\mathbf{u}}\right)+\left(\mathbf{K}_{\text {neum }} \mathbf{d}-\overline{\mathbf{t}}\right)^{T} \mathbb{A}_{\text {neum }}\left(\mathbf{K}_{\text {neum }} \mathbf{d}-\overline{\mathbf{t}}\right) \tag{7.23}
\end{align*}
$$

where $\mathbb{A}_{\text {inc }}=\alpha_{\text {inc }} \mathbf{I}$ is a square diagonal matrix collecting the weights associated to the discrete incompressibility equations, $\mathbb{A}_{\text {dir }}=\alpha_{d i r} \mathbf{I}$ collects the weights associated to Dirichlet boundary conditions, and $\mathbb{A}_{\text {neum }}=\alpha_{\text {neum }} \mathbf{I}$ collects the weights associated to Neumann boundary conditions. The total weighted error can be finally rewritten as

$$
\begin{equation*}
E_{w}=(\mathbf{K d}-\mathbf{f})^{T} \mathbb{A}(\mathbf{K} \mathbf{d}-\mathbf{f}) \tag{7.24}
\end{equation*}
$$

where $\mathbb{A}$ is a diagonal matrix with expression

$$
\mathbb{A}=\left[\begin{array}{cccc}
\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0}  \tag{7.25}\\
\mathbf{0} & \alpha_{\text {inc }} \mathbf{I} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \alpha_{\text {dir }} \mathbf{I} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \alpha_{\text {neum }} \mathbf{I}
\end{array}\right]
$$

The weighted error (7.24) is then minimized with respect to the control nodal variables d, therefore

$$
\begin{equation*}
\frac{\partial E_{w}}{\partial \mathbf{d}}=0 \tag{7.26}
\end{equation*}
$$

which implies that

$$
\begin{equation*}
\mathbf{K}^{T} \mathbb{A} \mathbf{K d}-\mathbf{K}^{T} \mathbb{A} \mathbf{f}=\mathbf{0} \tag{7.27}
\end{equation*}
$$

The matrix $\tilde{\mathbf{K}}=\mathbf{K}^{T} \mathbb{A} \mathbf{K}$ is a square symmetric matrix, and can be inverted using suitable algorithms, alleviating the computational cost of the method.

### 7.2.2 Choice of the weights

The choice of the weights to be imposed on Equation (7.24) is an important topic for the application of LSRM, since a wrong definition of weights may prevent the convergence of the numerical method.

The rigorous analysis conducted by Chi et al. (2014), which takes in account the particular choice of the shape functions (in that case Radial Basis Functions are used), leads to the selection of the following weights

$$
\begin{equation*}
\alpha_{\text {inc }}=\left(\mu N_{S}\right)^{2} \quad \alpha_{\text {dir }}=\left(\mu N_{S}\right)^{2} \quad \alpha_{\text {neum }}=1 \tag{7.28}
\end{equation*}
$$

In the present work we prefer a different approach: in fact we base our analysis on the consideration that different equations have different scales, and therefore they contribute differently to the global error. In particular:

1. The equation of equilibrium has the dimensions of $\mu \partial^{2} u / \partial x^{2}$
2. The equation of incompressibility has the dimensions of $\partial u / \partial x$
3. The Dirichlet boundary conditions have the dimensions of $u$
4. The Neumann boundary conditions have the dimensions of $\mu \partial u / \partial x$

The terms collected in the matrix $\mathbf{D}_{x x}$ are proportional to $1 / h^{2}$, where $h$ is the distance between control nodes on a regular distribution; the terms collected in the discrete operator $\mathbf{D}_{x}$ are proportional to $1 / h$; furthermore we consider that the distance $h$ between two near control nodes is related to the total number $N_{S}$ of control nodes, in particular we can assume $h \simeq 1 / \sqrt{N_{S}}$. Hence we can write the correct scale of each equation in the form:

1. Equilibrium: $o\left(\mu / h^{2}\right)=o\left(\mu N_{S}\right)$
2. Incompressibility constraint: $o(1 / h)=o\left(\sqrt{N_{S}}\right)$
3. Dirichlet boundary conditions: $o(1)$
4. Neumann boundary conditions: $o(\mu / h)=o\left(\mu \sqrt{N_{S}}\right)$

In order to balance the weighted squared error in Equation (7.24), all components are requested to have at least the same dimensions, that are the ones of the squared equation of equilibrium, $\left(\mu N_{S}\right)^{2}$. The other weights, following this principle, are:

$$
\left\{\begin{array}{l}
\alpha_{\text {inc }}=C_{1} N_{S}  \tag{7.29}\\
\alpha_{\text {dir }}=C_{2} N_{S}^{2} \\
\alpha_{\text {neum }}=C_{3} N_{S}
\end{array}\right.
$$

where $C_{1}, C_{2}$, and $C_{3}$ are constants that, in a first approximation, we can consider unitary.

### 7.3 Solution of the Stokes problem

In the present section we apply the procedure introduced in Section 3 on some benchmarks governed by the Stokes equations. In particular we first solve the Stokes problem on a square domain with known analytical solution, and then on a quarter of annulus under body loads. For both problems we discuss the choice of the weights, as well as the most correct way of distributing the control nodes with respect to the collocation points.

### 7.3.1 Square with known analytical solution

In the following we solve the Stokes flow in a square in the domain $[-1,1] \mathrm{x}[-1,1]$ with the following exact solution

$$
\left\{\begin{array}{l}
u(x, y)=20 x y^{3}  \tag{7.30}\\
v(x, y)=5\left(x^{4}-y^{4}\right) \\
p(x, y)=60 x^{2} y-20 y^{3}+C
\end{array}\right.
$$

The problem is subjected to Dirichlet boundary conditions in accordance to Equation (7.30) on the whole domain boundary. The viscosity is set as $\mu=1$. The numerical solution is obtained using the Modified Finite Particle Method for spatial discretization and the Least Square Residual Method for the approximation of the resulting linear system. The selected weights for the error balance are $\alpha_{d i r}=N_{S}^{2}, \alpha_{\text {inc }}=N_{S}$. The weights $\alpha_{\text {neum }}$ are not defined in this case, since no Neumann boundary conditions are imposed on the present test.

The first numerical test is performed using a regular distribution both of collocation nodes and control nodes. In Figure 7.1 we show the convergence diagram of the 2 nd norm relative error, which is defined as

$$
\begin{equation*}
\text { relative error }=\frac{\left.\sqrt{\sum_{i=1}^{N_{C}}\left(u_{n u m, i}-u_{a n, ~} i\right.}\right)^{2}}{\sqrt{\sum_{i=1}^{N_{C}}\left(u_{a n, i}\right)^{2}}} \tag{7.31}
\end{equation*}
$$

Such a convergence diagram is given in terms of the number of control nodes, that is directly proportional to the dimensions of the matrix to invert, and thus, to the computational effort. In particular we notice the expected second-order slope of the error curve.

In Figure 7.2 we show the pressure distribution obtained combining MFPM and LSRM and remark the smoothness of the solution.


Figure 7.1: Square with exact solution (7.30): convergence diagram of the error using MFPM and LSRM


Figure 7.2: Square with exact solution (7.30): pressure distribution obtained with the MFPM combined with LSRM, using 58081 field nodes and 231361 collocation points.

Effects of random distributions of collocation points. Now we explore the effects of random distributions of collocation points: the problem with exact solution (7.30) is solved on the same geometry and with the same boundary conditions, using three different extremely random distributions of collocation points within the problem domain, and using a regular distribution of control nodes. For each test we use a constant ratio between the number of collocation points and the number of control nodes, that is $N_{C} / N_{S}=4$. From Figure 7.3 we notice that the errors computed with the different distributions of collocation points is extremely stable, and that the average of the errors follows a slope superior to the expected second order.


Figure 7.3: Square with exact solution (7.30): convergence diagram of the error in space using the MFPM combined with the LSRM, with different set of randomly distributed collocation points

Effects of the ratio between the number of control nodes and collocation points. In order to assess the optimal ratio between the number of control and collocation points we solve again the problem with exact solution (7.30) using, for each distribution of control nodes, different number of collocation points. The results in term of relative error are reported in Figure 7.4 (velocity field) and in Figure 7.5 (pressure field), where different lines correspond to different amounts of control nodes. In particular we notice from Figure 7.5 that the error of the pressure field is high when the ratio $N_{S} / N_{C} \rightarrow 1$, due to the violation of the inf-sup condition, whereas an optimal error is got when $\sqrt{N_{S} / N_{C}} \simeq 0.5$.

Effects of weights. The suitability of the weights imposed for the error balance is tested on a problem with the same geometry and exact solution (7.30), but different boundary conditions: in the following test, in fact, Dirichlet boundary conditions are imposed on the top and lower sides of the square, and conditions over the stress are imposed on the left and right sides.


Figure 7.4: Square with exact solution (7.30): relative error versus the ratio between collocation points and field nodes


Figure 7.5: Error of the pressure for the problem with exact solution (7.30) and different ratio between field nodes and collocation points

We investigate different values of the weights, with the following general expression

$$
\left\{\begin{array}{l}
\alpha_{i n c}=\mu N_{S}^{p}  \tag{7.32}\\
\alpha_{D}=\mu N_{S}^{q} \\
\alpha_{N}=N_{S}^{r}
\end{array}\right.
$$

where $p, q$, and $r$ are positive parameters, then we compute the 2 nd norm of the error using the following combinations of parameters:
(a) $p=1, q=1, r=0$
(b) $p=1, q=1, r=1$
(c) $p=1, q=2, r=1$
(d) $p=1, q=2.5, r=1$


Figure 7.6: Square clamped an all its edges: convergence diagram of the error with Neumann boundary conditions, using a Weighted Least Square Residual Method and MFPM discretization, for different sets of weights

In Figure 7.6 we show the convergence diagrams of the error for the investigated sets of parameters $p, q$, and $r$, and observe that the best results in terms of global error are achieved when $\alpha_{\text {inc }}=\mu N_{S}, \alpha_{d i r}=\mu N_{S}^{2}$, and $\alpha_{\text {neum }}=N_{S}$, that is what we expected from the theoretical analysis of Section 3.

### 7.3.2 Quarter of annulus under body load

In the following we apply the combination of MFPM and LSRM on the problem of a flow in a quarter of annulus (see Figure 7.7) with geometrical parameters $r=1$ and $R=4$. The fluid viscosity is set as $\mu=1$. The problem has been studied in Auricchio et al. (2007) using isogeometric shape functions. The analytical solution of this problem is

$$
\left\{\begin{array}{l}
u=10^{-6} x^{2} y^{4}\left(x^{2}+y^{2}-16\right)\left(x^{2}+y^{2}-1\right)\left(5 x^{4}+18 x^{2} y^{2}-85 x^{2}+13 y^{4}+80-153 y^{2}\right)  \tag{7.33}\\
v=-2 \cdot 10^{-6} x y^{5}\left(x^{2}+y^{2}-16\right)\left(x^{2}+y^{2}-1\right)\left(5 x^{4}-51 x^{2}+6 x^{2} y^{2}-17 y^{2}+16+y^{4}\right)
\end{array}\right.
$$

The internal body loads are computed from the analytical solution (7.33). Dirichlet boundary conditions are imposed on the whole domain boundary.


Figure 7.7: Quarter of annulus under body loads: geometry and boundary conditions
For the solution of the present problem we use regular distributions of collocation points on a cylindrical reference frame, and regular distributions of control nodes on a Cartesian equispaced grid (an example of such distributions is reported in Figure 7.8). In Figure 7.9 we show the convergence diagram of the 2 nd norm relative error obtained for different numbers of control nodes and collocation nodes: we notice the expected second-order accuracy of the method.

Effects of the ratio between the number of control nodes and collocation points. We also investigate the effects of the ratio between the number of control nodes and the number collocation points. In Figure 7.10 we show the relative error versus the ratio $\sqrt{N_{S} / N_{C}}$. Different lines correspond to different amounts of control nodes. We notice that in all cases we obtain high errors for $N_{S} / N_{C} \rightarrow 1$, due to the violation of the inf-sup condition, whereas an optimal relative error is obtained when $\sqrt{N_{S} / N_{C}} \simeq 0.5$.

Effects of random distributions of collocation points. Further analyses are performed to investigate the effects of random distributions of collocation points, and keeping uniform distributions of control nodes (an example of collocation and control points distribu-


Figure 7.8: Quarter of annulus under body loads: regular distribution of 441 collocation and 83 control nodes


Figure 7.9: Quarter of annulus under body loads: convergence diagram of the error using MFPM and LSRM


Figure 7.10: Quarter of annulus under body loads: relative error versus the ratio between the number of field nodes and collocation points
tions are reported in Figure 7.11). In particular, for each distribution of control nodes, three different random distributions of collocation nodes are tested, and the errors are reported in Figure 7.12, where the dotted line correspond to the average error obtained in the tests. We notice that the error is extremely stable for the different distributions and its average follows the expected second-order convergence.

### 7.4 Navier-Stokes Equations

In the present section we solve the Navier-Stokes equations (7.14), which present a further difficulty with respect to the Stokes equations, since they are non-linear equations and therefore a proper procedure for handling the non-linearity is needed.

In the following we restrict to the stationary case, therefore equations (7.14) are modified in the form

$$
\left\{\begin{array}{l}
\rho \mathbf{u} \cdot \nabla \mathbf{u}+\nabla p=\mu \Delta \mathbf{u}+\mathbf{b}  \tag{7.34}\\
\nabla \cdot \mathbf{u}=0
\end{array}\right.
$$

System (7.34) can be rewritten highlighting the convective velocity $\mathbf{u}_{c}$ :

$$
\left\{\begin{array}{l}
\rho \mathbf{u}_{c} \cdot \nabla \mathbf{u}+\nabla p=\mu \Delta \mathbf{u}+\mathbf{b}  \tag{7.35}\\
\nabla \cdot \mathbf{u}=0
\end{array}\right.
$$

Systems (7.34) and (7.35) coincide when the convective velocity is chosen as $\mathbf{u}_{c}=\mathbf{u}$.


Figure 7.11: Quarter of annulus under body loads: random distribution of 441 collocation points and regular distribution of 83 control nodes


Figure 7.12: Quarter of annulus under body loads: convergence diagram of the error for different random distributions of collocation points

The solution procedure for the Navier-Stokes equation consists in an iterative linearization procedure, as usual for non-linear equations. In general for Navier-Stokes equations two different linearization procedures are used:

- Picard linearization: the convective velocity at the iteration $k$ is chosen as $\mathbf{u}_{c}=\mathbf{u}^{k}$; therefore the MFPM spatial discretization of System (7.35) is

$$
\left\{\begin{array}{l}
\rho \mathbf{I} \hat{\mathbf{u}}^{k} \cdot \mathbf{D}_{x} \hat{\mathbf{u}}^{k+1}+\rho \mathbf{I} \hat{\mathbf{v}}^{k} \cdot \mathbf{D}_{y} \hat{\mathbf{u}}^{k+1}+\mathbf{D}_{x} \hat{\mathbf{p}}^{k+1}=\mu\left(\mathbf{D}_{x x}+\mathbf{D}_{y y}\right) \hat{\mathbf{u}}^{k+1}+\mathbf{b}_{x}  \tag{7.36}\\
\rho \mathbb{I} \hat{\mathbf{u}}^{k} \cdot \mathbf{D}_{x} \hat{\mathbf{v}}^{k+1}+\rho \mathbf{I} \hat{\mathbf{v}}^{k} \cdot \mathbf{D}_{y} \hat{\mathbf{v}}^{k+1}+\mathbf{D}_{y} \hat{\mathbf{p}}^{k+1}=\mu\left(\mathbf{D}_{x x}+\mathbf{D}_{y y}\right) \hat{\mathbf{v}}^{k+1}+\mathbf{b}_{y} \\
\mathbf{D}_{x} \hat{\mathbf{u}}^{k+1}+\mathbf{D}_{y} \hat{\mathbf{v}}^{k+1}=0
\end{array}\right.
$$

that reads, after linearization:

$$
\left[\begin{array}{ccc}
\rho \mathbf{A}-\mu \mathbf{L} & \mathbf{0} & \mathbf{D}_{x}  \tag{7.37}\\
\mathbf{0} & \rho \mathbf{A}-\mu \mathbf{L} & \mathbf{D}_{y} \\
\mathbf{D}_{x} & \mathbf{D}_{y} & \mathbf{0}
\end{array}\right]\left(\begin{array}{c}
\Delta \hat{\mathbf{u}}^{k+1} \\
\Delta \hat{\mathbf{v}}^{k+1} \\
\Delta \hat{\mathbf{p}}^{k+1}
\end{array}\right)=\mathbf{R}^{k}
$$

where

$$
\left\{\begin{array}{l}
\mathbf{A}=d(\mathbb{I} \hat{\mathbf{u}}) \mathbf{D}_{x}+d(\mathbb{I} \hat{\mathbf{v}}) \mathbf{D}_{y}  \tag{7.38}\\
\mathbf{L}=\mathbf{D}_{x x}+\mathbf{D}_{y y}
\end{array}\right.
$$

and $\mathbf{R}^{k}$ is the residual of the equation. In Equation (7.38) we denote with $d(\mathbf{q})$ a square diagonal matrix whose elements are the elements of the generic vector $\mathbf{q}$.

- Newton-Raphson linearization: in the second case the convective velocity at the iteration $k$ is chosen as $\mathbf{u}_{c}=\mathbf{u}^{k+1}$; therefore the MFPM spatial discretization of System (7.35) reads

$$
\left\{\begin{array}{l}
\rho \mathbf{I} \hat{\mathbf{u}}^{k+1} \cdot \mathbf{D}_{x} \hat{\mathbf{u}}^{k+1}+\rho \mathbf{I} \hat{\mathbf{v}}^{k} \cdot \mathbf{D}_{y} \hat{\mathbf{u}}^{k+1}+\mathbf{D}_{x} \hat{\mathbf{p}}^{k+1}=\mu\left(\mathbf{D}_{x x}+\mathbf{D}_{y y}\right) \hat{\mathbf{u}}^{k+1}+\mathbf{b}_{x}  \tag{7.39}\\
\rho \mathbf{I} \hat{\mathbf{u}}^{k+1} \cdot \mathbf{D}_{x} \hat{\mathbf{v}}^{k+1}+\rho \mathbf{I} \hat{\mathbf{v}}^{k} \cdot \mathbf{D}_{y} \hat{\mathbf{v}}^{k+1}+\mathbf{D}_{y} \hat{\mathbf{p}}^{k+1}=\mu\left(\mathbf{D}_{x x}+\mathbf{D}_{y y}\right) \hat{\mathbf{v}}^{k+1}+\mathbf{b}_{y} \\
\mathbf{D}_{x} \hat{\mathbf{u}}^{k+1}+\mathbf{D}_{y} \hat{\mathbf{v}}^{k+1}=0
\end{array}\right.
$$

and therefore the linearized system reads

$$
\left[\begin{array}{ccc}
-\mu \mathbf{L}+\rho \mathbf{N L}_{u u} & \rho \mathbf{N} \mathbf{L}_{u v} & \mathbf{D}_{x}  \tag{7.40}\\
\rho \mathbf{N L}_{v u} & -\mu \mathbf{L}+\rho \mathbf{N L}_{v v} & \mathbf{D}_{y} \\
\mathbf{D}_{x} & \mathbf{D}_{y} & \mathbf{0}
\end{array}\right]\left(\begin{array}{c}
\Delta \hat{\mathbf{u}}^{k+1} \\
\Delta \hat{\mathbf{v}}^{k+1} \\
\Delta \hat{\mathbf{p}}^{k+1}
\end{array}\right)=\mathbf{R}^{k}
$$

where

$$
\left\{\begin{array}{l}
\mathbf{N L}_{u u}=\mathbf{A}+d\left(\mathbf{D}_{x} \hat{\mathbf{u}}^{k}\right) \mathbb{I}  \tag{7.41}\\
\mathbf{N L}_{u v}=d\left(\mathbf{D}_{y} \hat{\mathbf{u}}^{k}\right) \mathbb{I} \\
\mathbf{N L}_{v u}=d\left(\mathbf{D}_{x} \hat{\mathbf{v}}^{k}\right) \mathbb{I} \\
\mathbf{N L}_{v v}=\mathbf{A}+d\left(\mathbf{D}_{y} \hat{\mathbf{v}}^{k}\right) \mathbb{I}
\end{array}\right.
$$

It is evident that the Newton-Raphson algorithm represents the most accurate linearization of Navier-Stokes equations (7.34), and therefore it shows faster rate of convergence: neverthe-
less it is shown in Elman et al. (2014) that such a linearization strategy needs an initial guess solution closer and closer to the converged solution as much as the Reynolds number $R e$ is high, where $R e=\rho L V / \mu$ is the ratio among the fluid inertial and viscous forces. $L$ and $V$ are characteristic length and velocity of the problem.

In the case of Picard linearization, conversely, the initial guess velocity can be chosen in a larger bubble, and therefore convergence is more easily reached, even if the rate of convergence is not optimal.

In both cases, at each iteration $k$, a linear system has to be solved. In both cases the general form is

$$
\begin{equation*}
\mathbf{K}^{k} \Delta \hat{\mathbf{d}}^{k+1}=\mathbf{R}^{k} \tag{7.42}
\end{equation*}
$$

where the superscript $k$ reminds that we are implementing an iterative process, and $\Delta \hat{\mathbf{d}}^{k+1}=$ $\left[\Delta \hat{\mathbf{u}}^{k+1} \Delta \hat{\mathbf{v}}^{k+1} \Delta \hat{\mathbf{p}}^{k+1}\right]^{T}$ is the increment of the nodal variables at iteration $k+1$.

### 7.4.1 Navier-Stokes equations and Least Square Residual Method

When collocation and control nodes coincide, the solution of system (7.42) with suitable boundary conditions is found by inversion of the matrix $\mathbf{K}^{k}$. Howewever this procedure, as in the linear case, can lead to problems of pressure instability, and therefore also in the non-linear case we use a LSRM approach for the solution of the problem.

The procedure consists in considering a number of collocation points higher than the number of control nodes, therefore at each iteration system (7.42) is overdetermined: therefore its solution can be approximated through an error minimization, following a procedure completely similar to the linear case; the weighted error at each iteration is computed as

$$
\begin{equation*}
E_{w}^{k}=\left(\mathbf{K} \Delta \hat{\mathbf{d}}^{k+1}-\mathbf{R}^{k}\right)^{T} \mathbb{A}^{k}\left(\mathbf{K} \Delta \hat{\mathbf{d}}^{k+1}-\mathbf{R}^{k}\right) \tag{7.43}
\end{equation*}
$$

where $\mathbb{A}^{k}$ is the diagonal matrix of the squared weights, with components

$$
\begin{equation*}
\mathbb{A}_{i i}^{k}=\left(\frac{3 N_{\text {supp }}}{\sum_{j=1}^{3 N_{\text {supp }}} K_{i j}}\right)^{2} \tag{7.44}
\end{equation*}
$$

computed in order to restore the same order of magnitude of the terms in all equations. $N_{\text {supp }}$ is the number of supporting nodes of the collocation points to which the $i$-th row of $\mathbf{K}^{k}$ is associated.

Finally, at each iteration the approximated solution $\Delta \hat{\mathbf{d}}^{k+1}$ is the minimizer of the error $E_{w}^{k}$. The procedure is repeated until convergence of the weighted error under a predetermined tolerance. In order to address the different iteration solutions to convergence, it is possible to divide the problem in a predetermined number of substeps, in which the external data (i.e., boundary conditions or the internal body loads) are gradually increased. For each substep, the converged solution of the previous substep is used as initial guess solution.

### 7.5 Solution of the Navier Stokes problem using MFPM combined with LSRM

In the following, we apply what has been discussed in the previous section to a square under polynomial body loads, with known analytical solution, and on the well-known benchmark of the lid-driven cavity, at different values of the Reynold number.

### 7.5.1 Flow in a square domain

In the following we study the flow on a square in the domain $[-1,1] \mathrm{x}[-1,1]$ with analytical solution

$$
\left\{\begin{array}{l}
u(x, y)=20 x y^{3}  \tag{7.45}\\
v(x, y)=5\left(x^{4}-y^{4}\right) \\
p(x, y)=60 x^{2} y-20 y^{3}+C
\end{array}\right.
$$

Dirichlet boundary conditions (on the whole boundary) and internal body loads are imposed according to Equation (7.45). The problem is solved using second order MFPM in combination with the Least Square Residual Method, using the iterative procedure proposed in the previous section. The material parameters are set as $\rho=1 \mathrm{~kg} / \mathrm{m}^{3}$ and $\mu=1 \mathrm{~kg} / \mathrm{ms}$. In Figure 7.13 we show the convergence diagram of the error for this problem, and highlight the expected second-order convergence.


Figure 7.13: Square with analytical solution (7.45) in the non-linear case: convergence diagram of the error.

### 7.5.2 The lid-driven cavity

The domain of the lid-driven cavity is a square of side $L=1 \mathrm{~m}$. On all boundary, Dirichlet boundary condition are assigned, in particular

$$
\begin{cases}\mathbf{u}=\mathbf{0} & \text { on the left, lower and right sides }  \tag{7.46}\\ \mathbf{u} \cdot \mathbf{n}=0 \text { and } \mathbf{u} \cdot \mathbf{t}=\bar{U} & \text { on the top side }\end{cases}
$$

where $\mathbf{n}$ and $\mathbf{t}$ are the normal and tangential unit vectors at the boundary, and $\bar{U}$ is the tangential velcoity at the top side of the domain, which has been set as $\bar{U}=1 \mathrm{~m} / \mathrm{s}$. We solve the problem with $\rho=1 \mathrm{~kg} / \mathrm{m}^{3}$ and two different values for the viscosity (i.e., $\mu=0.0025 \mathrm{~kg} / \mathrm{ms}$ and $\mu=0.0001 \mathrm{~kg} / \mathrm{ms}$ ) corresponding to the Reynolds numbers $R e=400$ and $R e=1000$. In Figures 7.14 and 7.15 we show our results obtained in terms of streamlines.


Figure 7.14: Lid-driven cavity problem $(R e=400)$ : streamlines solution using MFPM and LSRM.
In absence of analytical solution, we compare our results with those obtained by Ghia et al. (1982) in terms of horizontal velocity profile at the middle vertical axis ( $x=0$ in our reference frame) for $R e=400$ (Figure 7.16) and $R e=1000$ (Figure 7.17). For both cases we notice a substantial agreement between our results and the reference solution.


Figure 7.15: Lid-driven cavity problem ( $R e=1000$ ): streamlines solution using MFPM and LSRM.


Figure 7.16: Lid-driven cavity problem $(R e=400)$ : velocity profile in the $x$-direction along the axis $x=0$ and comparison with the reference solution by Ghia et al. (1982).


Figure 7.17: Lid-driven cavity problem ( $R e=1000$ ): velocity profile in the $x$-direction along the axis $x=0$ and comparison with the reference solution by Ghia et al. (1982).

## Chapter 8

## Conclusions

The topic of the present thesis is the study and the development of the Modified Finite Particle Method (MFPM) from its first introduction in the scientific literature (Asprone et al., 2010) until its last developments already unpublished.

In the present work we deeply analyze and review the original MFPM formulation and study its characteristics. In particular we underline the computational costs connected with the original formulation, where integrals had to be computed as necessary step for derivative approximation. Such a need has been removed introducing a MFPM novel formulation, where the projection is performed among vectors and not of functions. In Chapter 3 it is evidenced the consistent time saving connected with these procedure, and also a reduction of the approximation error.

Both original and novel formulations are then compared with existing meshless collocation methods available in the literature, in particular with the Generalized Finite Difference Method, and applied to linear elasticity, both in statics and in dynamics. The results in terms of error slope are shown in the thesis and the expected second order consistency of the method is confirmed.

At a later stage MFPM has been tested on the incompressible elasticity equations. In this case we notice that a simple discretization of the variables involved (displacements and pressure) do not respect the restriction imposed by the inf-sup (or $L B B$ ) condition, and therefore the pressure field exhibits unphysical obscillations. The way of overcoming this difficulty are essentially two: the first one consists in introducing a different set of equations, and this is what has been done by many authors; the second strategy is to slightly modify the numerical approximation method, and use the original incompressibility equations.

Concerning the first strategy, we apply the Modified Finite Particle Method to five different formulations presented in the literature, and verify that only some of them actually respect the incompressibility constraint. When the constraint is respect, however, a correct second order accuracy is achieved in the numerical tests.

The second strategy for dealing with the problem of incompressible solids and fluids is to use the Modified Finite Particle Method in conjunction with a Least Squares Residual Method, following the idea introduced by Hu et al. (2007) and (Chi et al., 2014) and implemented using the Radial Basis Functions collocation method. Such a conjunction however requires a modification of the Modified Finite Particle Method, that here we call extended MFPM,
that shows higher robustness with respect to last versions when dealing with unstructured distributions of collocation points. The method also avoids spurious oscillations of the pressure field, and then in this case the original incompressibility equations have been used. The last version of MFPM has been finally used for the solution of flows in an Eulerian point of view, that is, the Navier-Stokes problem has been solved. Such problem is modelled by non-linear equations, and then the MFPM has been extended to the non-linear case. The obtained results show again the robustness of the method and the expected accuracy.

The current version of the Modified Finite Particle Method permits satisfactory approximations of compressible and incompressible elasticity problems, in particular it does not suffer from numerical pathologies typical of incompressibility, and it is particularly robust with respect to random distribution of particles; nevertheless the method still needs further investigations in order to properly set some parameters for the improvement of the method efficiency. The methods need to be properly developed in order to be efficiently applied to Lagrangian fluid-dynamics, that is the most natural application of every meshless method.

## Bibliography

Arroyo, M. and M. Ortiz (2006). Local max-ent approximation schemes: A seamless bridge between finite elements and meshfree methods. International Journal for Numerical Methods in Engineering 65, 2167-2202.

Arroyo, M. and M. Ortiz (2007). Local Maximum-Entropy Approximation Schemes. In Meshfree Methods for Partial Differential Equations III. Springer Berlin Heidelberg.

Asprone, D., F. Auricchio, G. Manfredi, A. Prota, A. Reali, and G. Sangalli (2010). Particle Methods for a 1d Elastic Model Problem: Error Analysis and Development of a SecondOrder Accurate Formulation. Computational Modeling in Engineering 63 Sciences 62, 1-21.

Asprone, D., F. Auricchio, A. Montanino, and A. Reali (2014). A modified finite particle method: Multi-dimensional elasto-statics and dynamics. International Journal for Numerical Methods in Engineering.

Asprone, D., F. Auricchio, A. Montanino, and A. Reali (2015). Review of the modified finite particle method and application to incompressible solids. The International Journal of Multiphysics 9(3), 235-248.

Asprone, D., F. Auricchio, and A. Reali (2011). Novel finite particle formulations based on projection methodologies. International Journal for Numerical Methods in Fluids 65, 13761388.

Asprone, D., F. Auricchio, and A. Reali (2014). Modified finite particle method: applications to elasticity and plasticity problems. International Journal of Computational Methods 11 (01).

Atluri, S. and T.-L. Zhu (2000). The meshless local petrov-galerkin (mlpg) approach for solving problems in elasto-statics. Computational Mechanics 25(2-3), 169-179.

Atluri, S. N. and T. Zhu (1998). A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. Computational Mechanics 22, 117-127.

Aurenhammer, F. (3 September 1991). Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. ACM Computing Survey 23, 345-405.

Auricchio, F., L. B. da Veiga, A. Buffa, C. Lovadina, A. Reali, and G. Sangalli (2007). A fully locking-free isogeometric approach for plane linear elasticity problems: a stream function formulation. Computer methods in applied mechanics and engineering 197(1), 160-172.

Auricchio, F., L. B. da Veiga, T. Hughes, A. Reali, and G. Sangalli (2012). Isogeometric collocation for elastostatics and explicit dynamics. Computer Methods in Applied Mechanics and Engineering 249-252(0), 2-14.

Bell, J. B., P. Colella, and H. M. Glaz (1989). A second-order projection method for the incompressible navier-stokes equations. Journal of Computational Physics 85(2), 257-283.

Benito, J., F. Ureña, and L. Gavete (2001). Influence of several factors in the generalized finite difference method. Applied Mathematica Modelling 25, 1039-1053.

Benito, J., F. Ureña, and L. Gavete (2007). Solving parabolic and hyperbolic equations by the generalized finitedifference method. Journal of computational and applied mathematics 209, 208-233.

Brezzi, F. (1974). On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique 8(R2), 129-151.

Brezzi, F. and J. Douglas Jr (1988). Stabilized mixed methods for the stokes problem. Numerische Mathematik 53(1-2), 225-235.

Brezzi, F. and M. Fortin (1991). Mixed and hybrid finite element methods. Springer-Verlag New York, Inc.

Brown, D. L., R. Cortez, and M. L. Minion (2001). Accurate projection methods for the incompressible navier-stokes equations. Journal of Computational Physics 168(2), 464499.

Buhmann, M. D. (2000). Radial basis functions. Acta Numerica 2000 9, 1-38.
Chen, J. K., J. E. Beraun, and T. C. Carney (1999). A corrective smoothed particle method for boundary value problems in heat conduction. International Journal for Numerical Methods in Engineering 46, 231-252.

Chen, J. K., J. E. Beraun, and C. J. Jih (1999). An improvement for tensile instability in smoothed particle hydrodynamics. Computational Mechanics 23, 279-287.

Chi, S.-W., J.-S. Chen, and H.-Y. Hu (2014). A weighted collocation on the strong form with mixed radial basis approximations for incompressible linear elasticity. Computational Mechanics 53(2), 309-324. cited By (since 1996)0.

Chorin, A. J. (1967). A numerical method for solving incompressible viscous flow problems. Journal of computational physics 2(1), 12-26.

Chorin, A. J. (1968). Numerical solution of the navier-stokes equations. Mathematics of computation 22(104), 745-762.

Cook, R. and J. Al-Abdulla (1969). Some Plane Quadrilateral 'Hybrid' Finite Element. American Institute of Aeronautics and Astronautics Journal 7, 2184-2185.

Cyron, C. J., M. Arroyo, and M. Ortiz (2009). Smooth, second order, non-negative meshfree approximants selected by maximum entropy. International Journal for Numerical Methods in Engineering 79, 1605-1632.

Ding, H., C. Shu, K. Yeo, and D. Xu (2004a). Development of least-square-based twodimensional finite-difference schemes and their application to simulate natural convection in a cavity. Computers 8 fluids $33(1), 137-154$.

Ding, H., C. Shu, K. Yeo, and D. Xu (2004b). Simulation of incompressible viscous flows past a circular cylinder by hybrid fd scheme and meshless least square-based finite difference method. Computer Methods in Applied Mechanics and Engineering 193(9), 727-744.
E, W. and J.-G. Liu (1995). Projection method i: convergence and numerical boundary layers. SIAM Journal on Numerical Analysis, 1017-1057.

E, W. and J.-G. Liu (2003). Gauge method for viscous incompressible flows. Communications in Mathematical Sciences 1 (2), 317-332.

Elman, H., D. Silvester, and A. Wathen (2014). Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Oxford University Press.

Franke, C. and R. Schaback (1998). Solving partial differential equations by collocation using radial basis functions. Applied Mathematics and Computation 93(1), 73-82.

Gavete, L., F. Ureña, J. Benito, and E. Salete (2012). A note on the dynamic analysis using the generalized finite difference method. Journal of Computational and Applied Mathematics 236, 3016-3025.

Ghia, U., K. N. Ghia, and C. Shin (1982). High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. Journal of computational physics 48 (3), 387-411.

Gingold, R. and J. Monaghan (1977). Smoothed Particle Hydrodynamics: theory and application to non spherical stars. Monthly Notices of the Royal Astronomical Society 181, 375-389.

Gresho, P. M. and R. L. Sani (1987). On pressure boundary conditions for the incompressible navier-stokes equations. International Journal for Numerical Methods in Fluids 7(10), 11111145.

Harlow, F. H., J. E. Welch, et al. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Physics of fluids 8(12), 2182.

Hon, Y. and Z. Yang (2009). Meshless collocation method by delta-shaped basis functions for default barrier model. Engineering Analysis with Boundary Elements 33(7), 951-958.

Hu, H., J. Chen, and W. Hu (2007). Weighted radial basis collocation method for boundary value problems. International Journal for Numerical Methods in Engineering 69(13), 27362757.

Johnson, G. R., R. A. Stryk, and S. R. Beissel (1996). SPH for high velocity impact computations. Computer Methods in Applied Mechanics and Engineering 139, 347 - 373.

Kansa, E. (1990a). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - II solutions to parabolic, hyperbolic and elliptic partial differential equations. Computers $\xi$ Mathematics with Applications 19, 147-161.

Kansa, E. J. (1990b). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics I surface approximations and partial derivative estimates. Computers 83 Mathematics with applications 19(8), 127-145.

Kim, J. and P. Moin (1985). Application of a fractional-step method to incompressible navierstokes equations. Journal of computational physics 59(2), 308-323.

LeVeque, R. J. (1955). Finite Difference Method for Ordinary and Partial DIfferential Equations, Chapter 1, pp. 10-11. SIAM.

Lin, H. and S. Atluri (2000). Meshless local petrov-galerkin(mlpg) method for convection diffusion problems. CMES(Computer Modelling in Engineering \& Sciences) 1(2), 45-60.

Lin, H. and S. Atluri (2001). The meshless local petrov-galerkin (mlpg) method for solving incompressible navier-stokes equations. CMES-Computer Modeling in Engineering and Sciences 2(2), 117-142.

Liu, W. K., S. Jun, S. Li, J. Adee, and T. Belytschko (1995). Reproducing Kernel Particle Methods for Structural Dynamics. International Journal for Numerical Methods in Engineering 38, 1655-1680.

Liu, W. K., S. Jun, and Y. F. Zhang (1995). Reproducing Kernel Particle Methods. International Journal for Numerical Methods in Fluids 20, 1081-1106.

Long, S. and S. Atluri (2002). A meshless local petrov-galerkin method for solving the bending problem of a thin plate. Computer Modeling in Engineering and Sciences 3(1), 53-64.

Lucy, L. (1977). A numerical approach to the testing of the fission hypothesis. The astronomical journal 82, 1013-1024.

Monaghan, J. and J. Lattanzio (1985). A refined particle method for astrophysical problems. Astronomy and Astrophysucs 149, 135-143.

Morris, J. P. (1996a). Analysis of Smoothed Particle Hydrodynamics with Applications. Ph. D. thesis, Department of Mathematics, Monash University.

Morris, J. P. (1996b). A study of the stability properties of SPH. Publications of The Astronomical Society of Australia 13, 97-102.

Quarteroni, A., F. Saleri, and A. Veneziani (2000). Factorization methods for the numerical approximation of navier-stokes equations. Computer methods in applied mechanics and engineering 188(1), 505-526.

Reutskiy, S. (2005). A boundary method of trefftz type for $\{$ PDEs $\}$ with scattered data. Engineering Analysis with Boundary Elements 29(7), 713-724.

Rosolen, A. and M. Arroyo (2013). Blending isogeometric analysis and local<i> maximum entropy</i> meshfree approximants. Computer Methods in Applied Mechanics and Engineering.

Rosolen, A., D. Millán, and M. Arroyo (2012). Second-order convex maximum entropy approximants with applications to high-order PDE. International Journal for Numerical Methods in Engineering, n/a-n/a.

Sani, R. L., J. Shen, O. Pironneau, and P. Gresho (2006). Pressure boundary condition for the time-dependent incompressible navier-stokes equations. International Journal for Numerical Methods in Fluids 50(6), 673-682.

Shannon, C. E. (2001). A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review 5(1), 3-55.

Strikwerda, J. C. (1984). Finite difference methods for the stokes and navier-stokes equations. SIAM Journal on Scientific and Statistical Computing 5(1), 56-68.

Ureña, F., J. Benito, and L. Gavete (2011). Application of the generalized finite difference method to solve the advection-diffusion equation. Journal of Computational and Applied Mathematics 2011, 1849-1855.

Ureña, F., E. Salete, J. Benito, and L. Gavete (2012). Solving third- and fourth-order partial differential equations using GFDM: application to solve paroblems of plates. International Journal of Computer Mathematics 89, 366-376.

Wang, C. and J.-G. Liu (2000). Convergence of gauge method for incompressible flow. Mathematics of computation 69(232), 1385-1407.

Wang, J. G. and G. R. Liu (2002). A point interpolation meshless method based on radial basis functions. International Journal for Numerical Methods in Engineering 54(11), 1623-1648.

Zhang, G. and R. Batra (2004). Modified smoothed particle hydrodynamics method and its application to transient problems. Computational Mechanics 34, 137-146.


[^0]:    ${ }^{1}$ We compute the spectral decomposition of the integral because $F(t)$ is a condition on the derivative of the solution, and not on the solution itself

